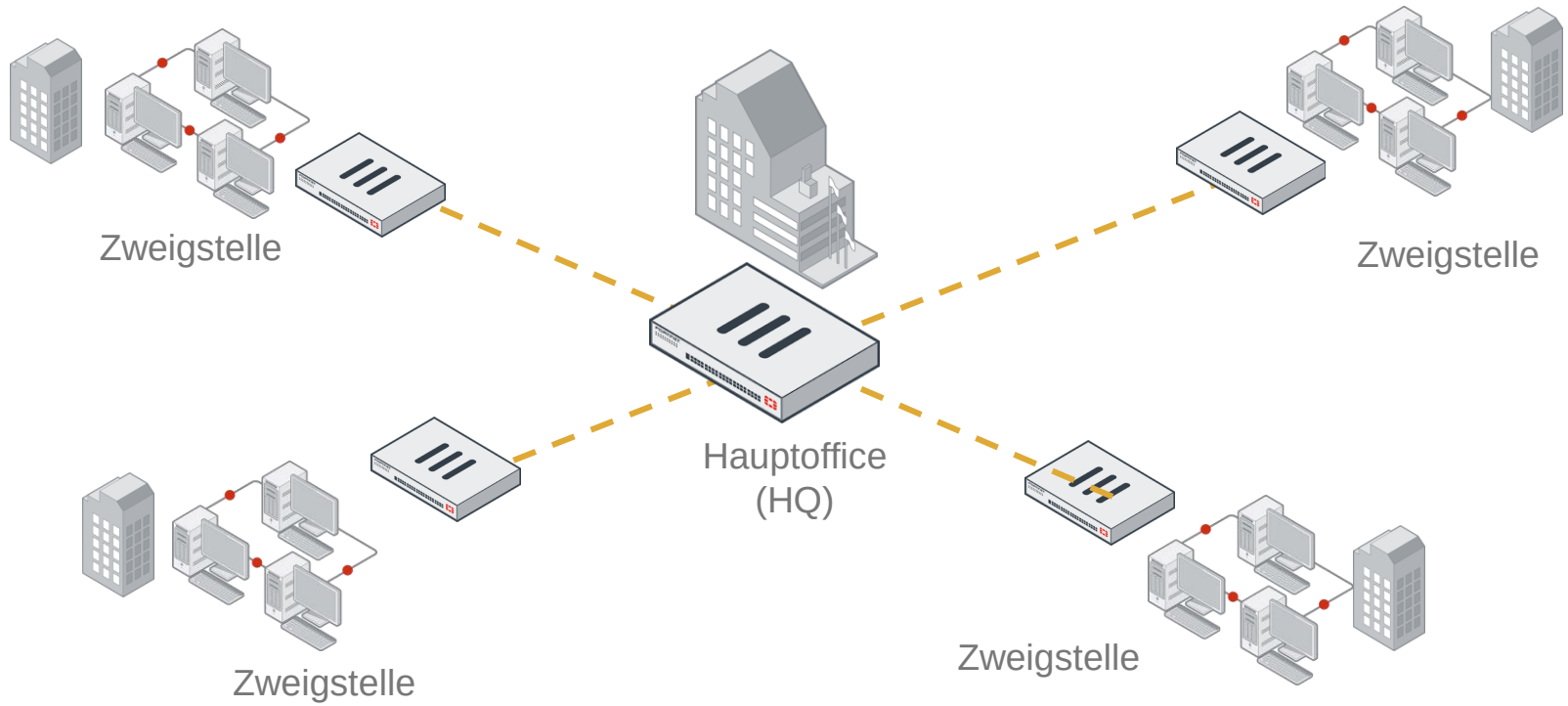
FortiOS 6.0 aus der Praxis - ADVPN

24. Oktober 2018 – Emmen

- Martin Ruesch, Technical Consultant, ALSO
- Gabriel Kaelin, Jonas Walker & Markus Frey, Systems Engineer, Fortinet

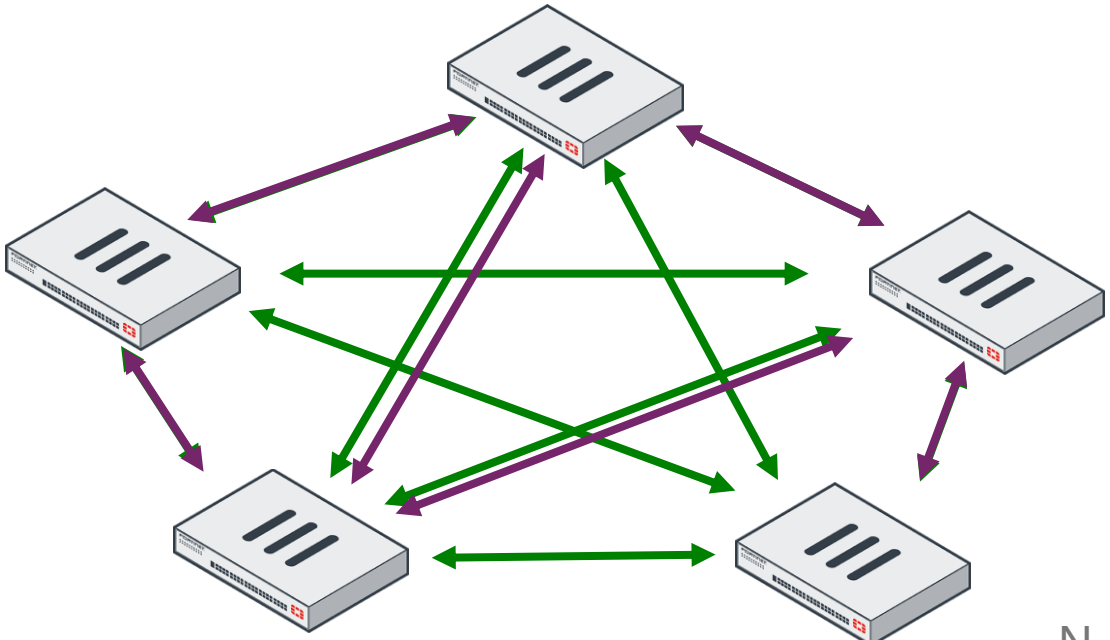
AD-VPN - Grundlagen

Hub-and-Spoke

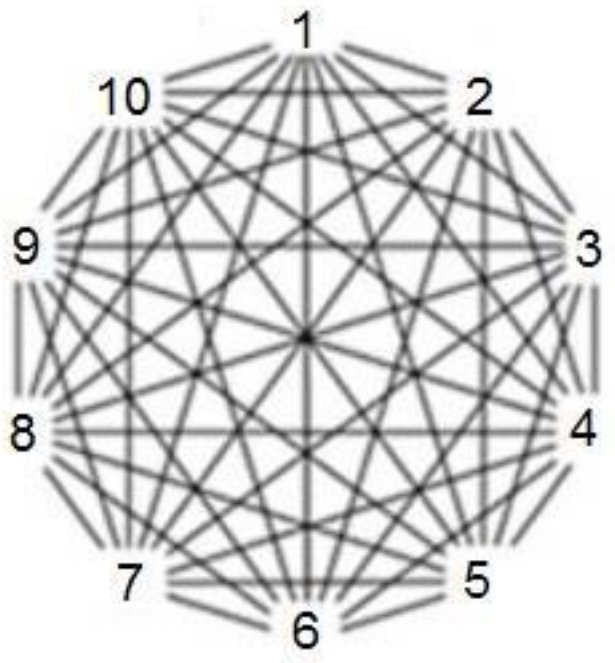


Statischer VPN Tunnel

Full Mesh und Partial Mesh



Full Mesh
Partial Mesh



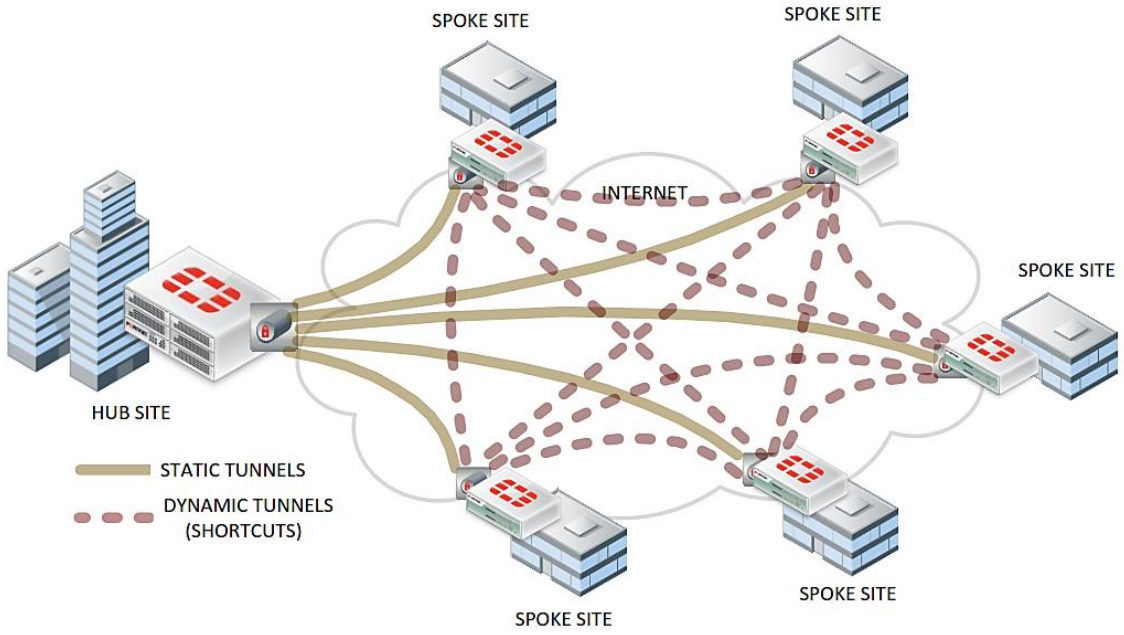
N Aussenstellen = N (N-1) / 2 Tunnels
10 Aussenstellen = 45 Tunnels !

Vergleich der VPN-Topologie

Hub-and-Spoke	Partial Mesh	Full Mesh
Einfache Konfiguration	Moderate Konfiguration	Komplexe Konfiguration
Wenig Tunnels	Mittlere Anzahl von Tunneln	Viele Tunnels
Zentral hohe Bandbreite	Mittlere Bandbreite auf der Hub Seite	Niedrige Bandbreite
Keine Störungstoleranz	Zum teil Störungs toleranz	Störungs Toleranz
Geringe Systemanforderungen an den Spokes, grosse Systemanforderung am Hub	Mittlere Systemanforderungen	Hohe Systemanforderungen
Skalierbar	Beschränkt skalierbar	Schwer zu skalieren
Keine direkte Kommunikation zwischen den Aussenstellen(Spokes)	Direkte Kommunikation zwischen einigen Aussenstellen(Spokes)	Direkte Kommunikation zwischen allen Aussenstellen (Spokes)

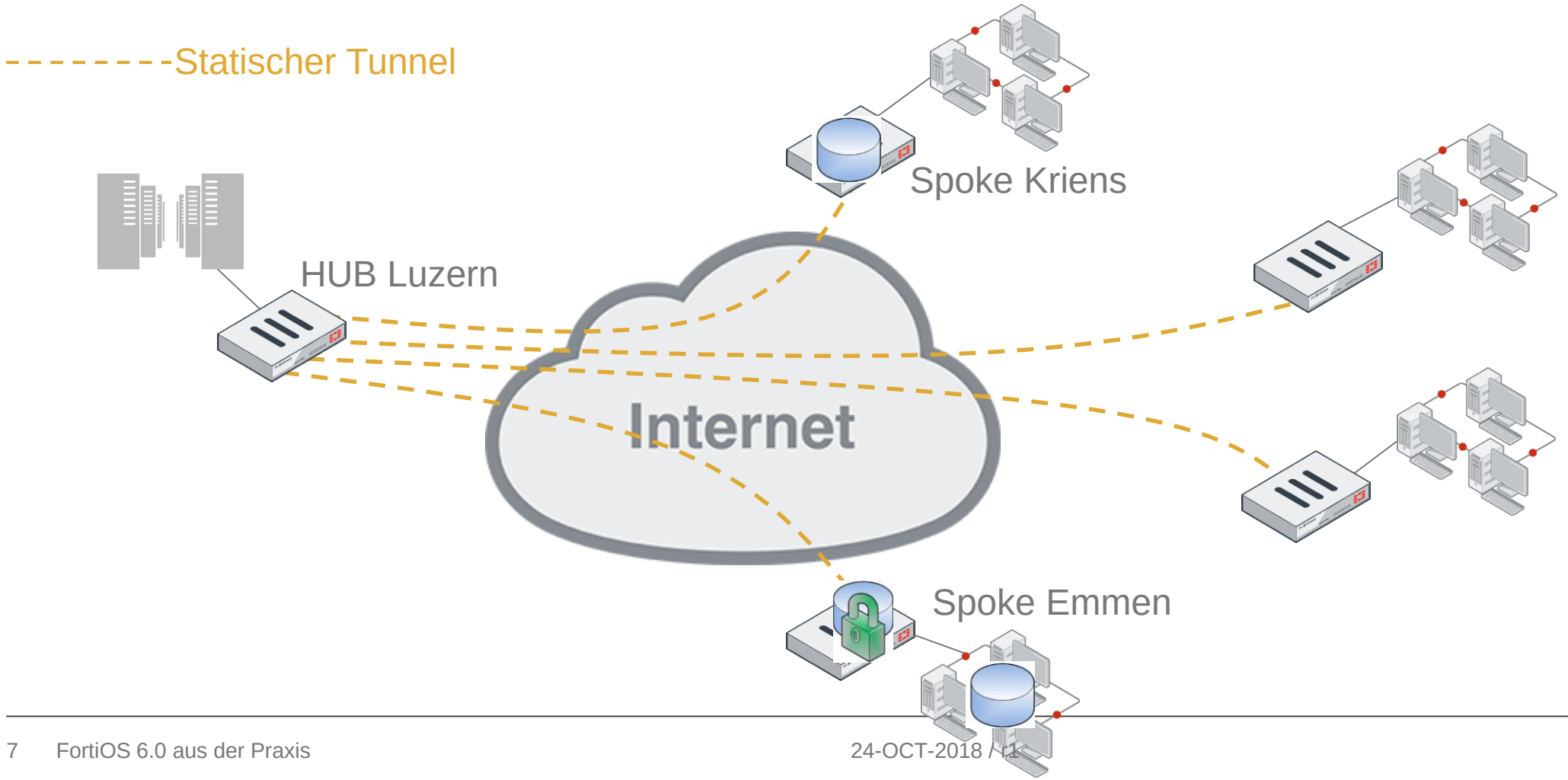
Auto Discovery VPN (ADVPN)

- Verhandelt dynamisch On-Demand direkte VPNs zwischen den Aussenstellen.
  - Bietet die Vorteile einer vollständigen Mesh-Topologie gegenüber einer Hub-and-Spoke- oder partiellen Mesh-Topologie.
  - Erfordert die Verwendung eines Routing-Protokolls für Aussenstellen, welches die Routen zu den anderen Spokes lernt. (dynamisches Routing)



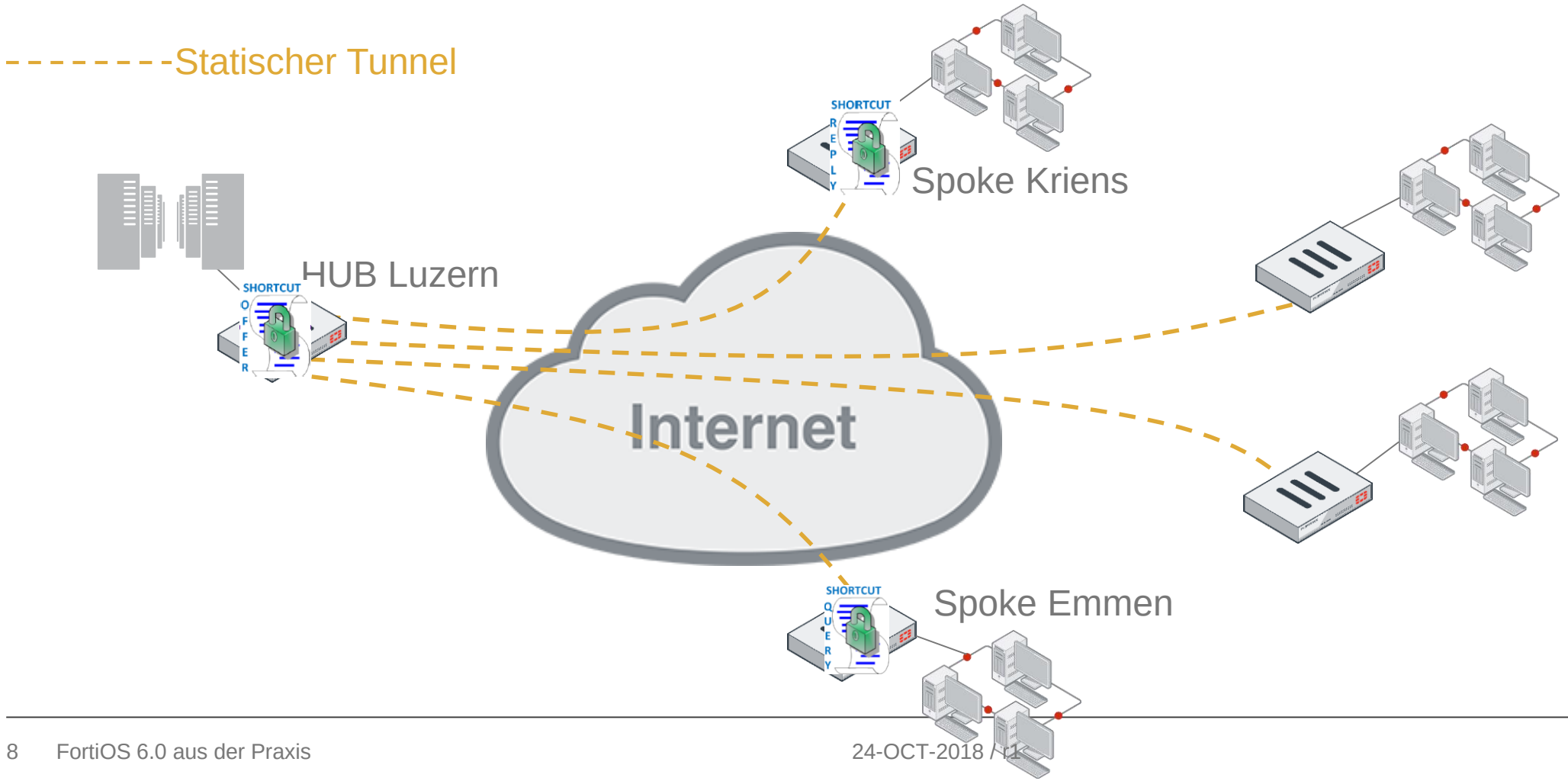
AD-VPN – Schritt 1

- 1. Shortcut wird initialisiert, wenn Daten durch den Hub fliessen.



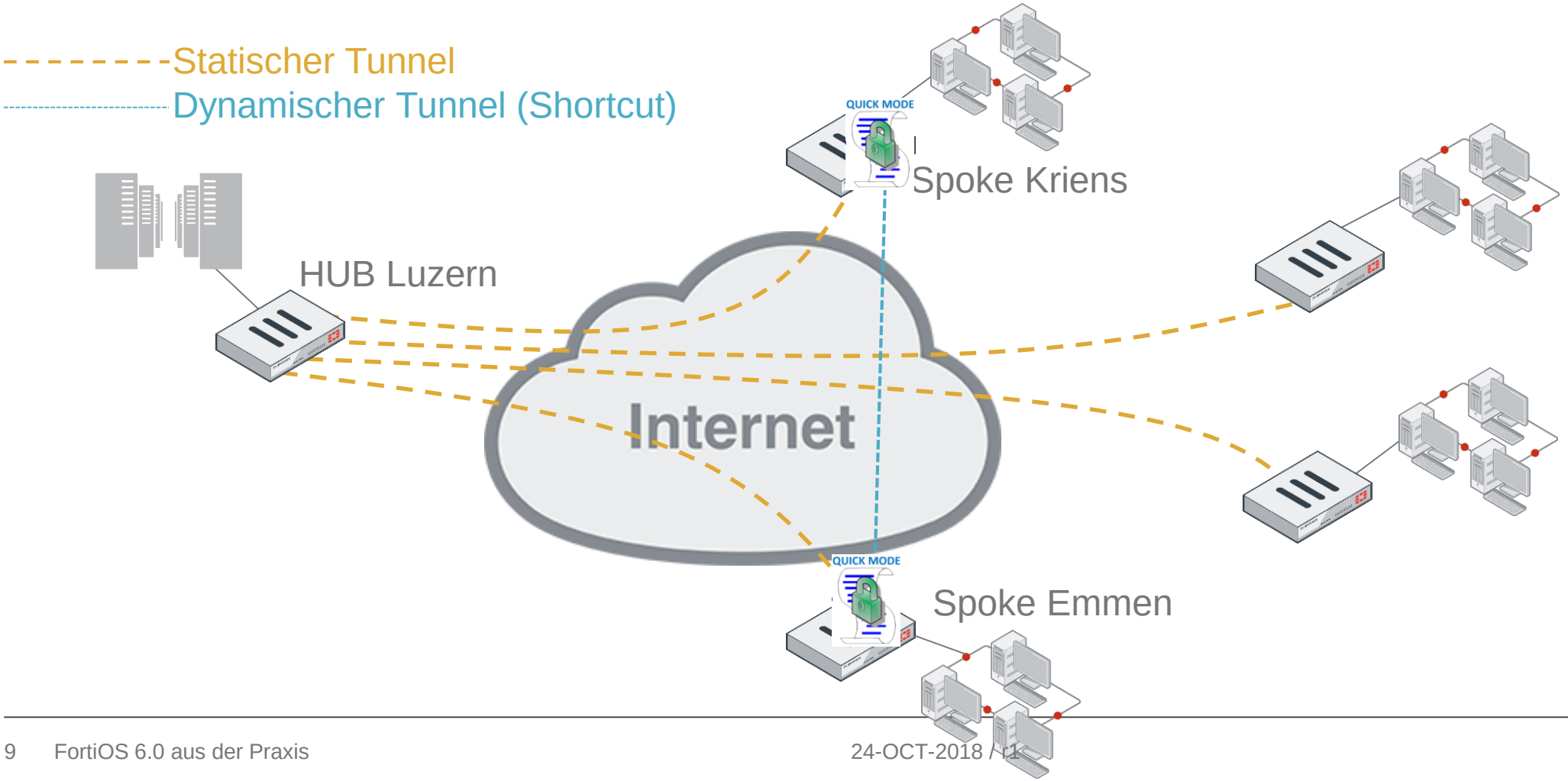
AD-VPN – Schritt 2

- 2. Die Aushandlung für den Shortcut wird durch den Hub organisiert.



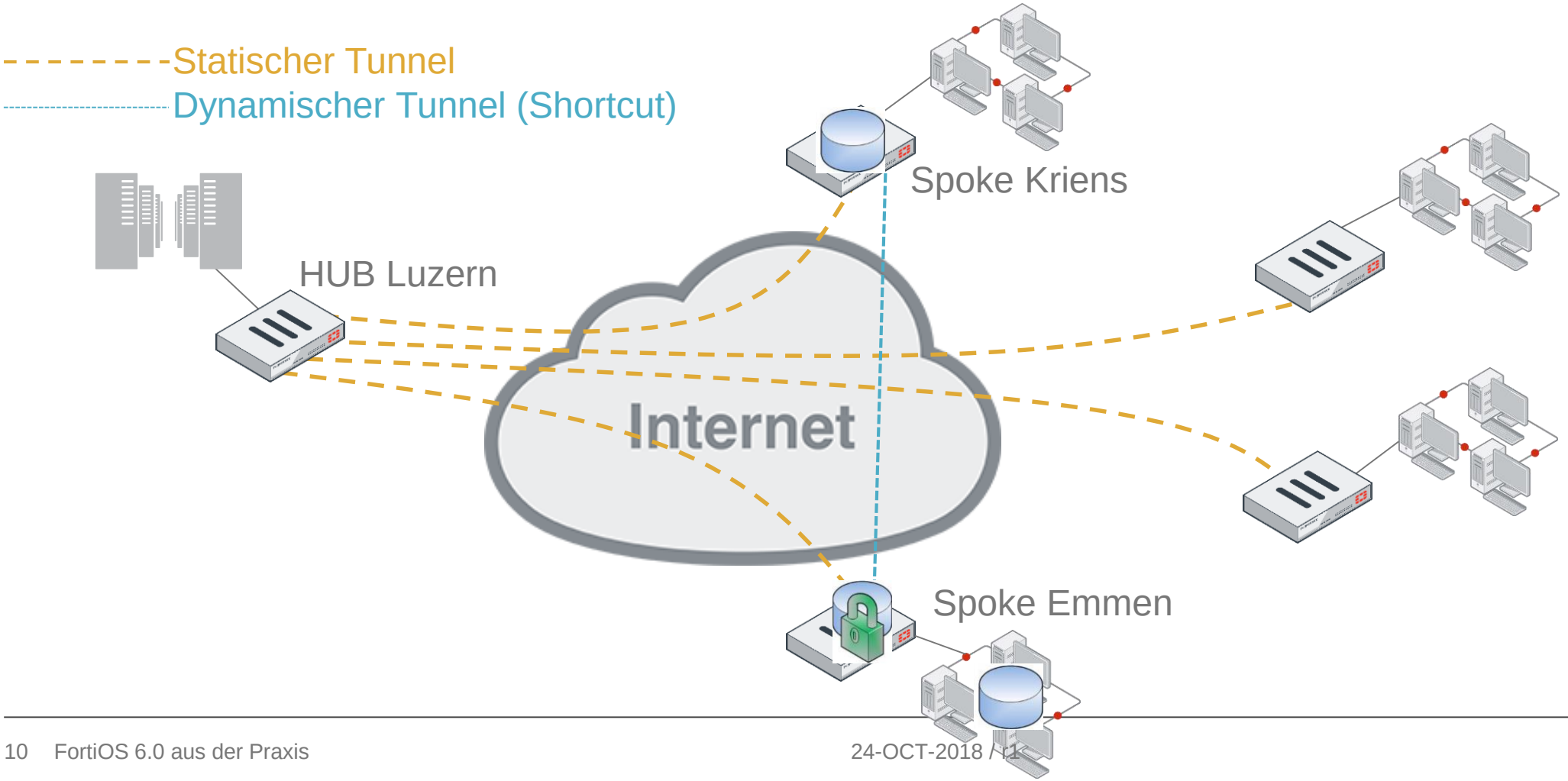
AD-VPN – Schritt 3

- 3. Der Shortcut wird zwischen den beiden Spokes eingerichtet.

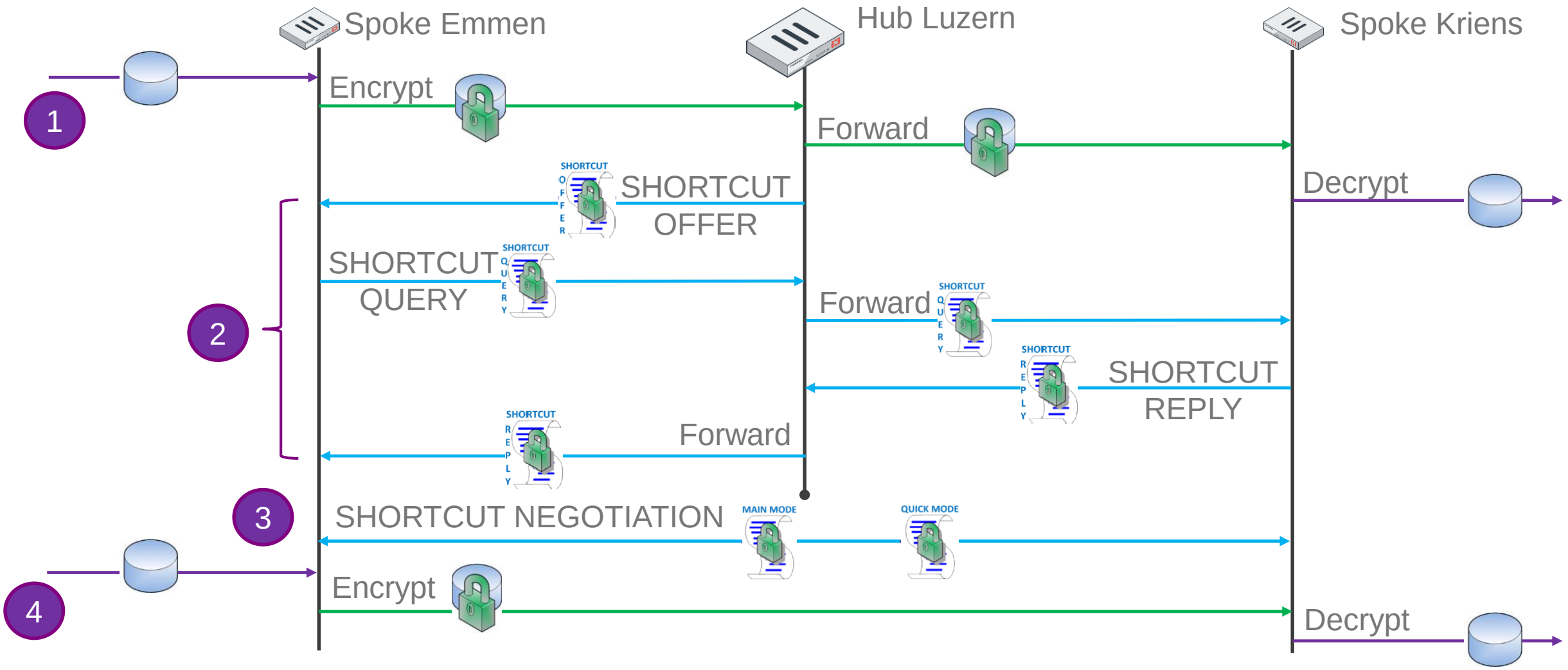


AD-VPN – Schritt 4

- 4. Der Spoke-to-Spoke Traffic fliesst nun durch den Shortcut.



Zusammenfassung – ADVPN Aufbau Shortcut

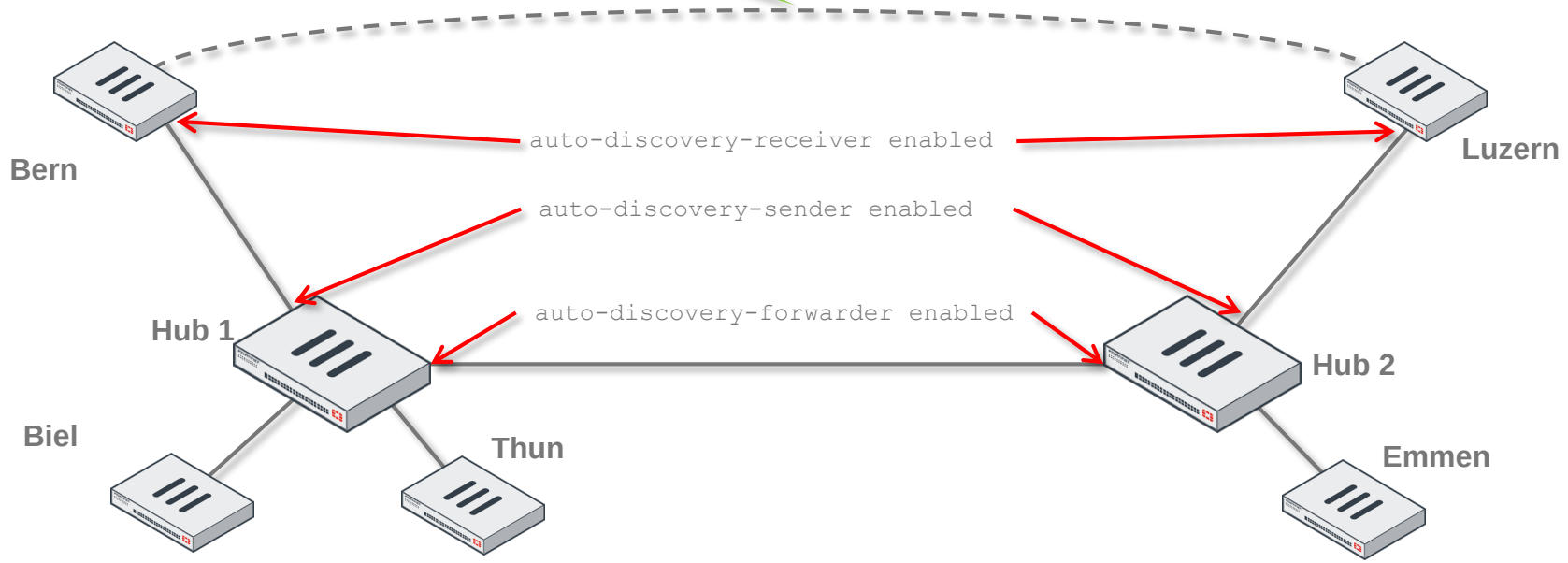


Fortinet Auto-Discovery VPN

- Fortinet ADVPN ist eine proprietäre Lösung, die ausschliesslich auf IKE & IPsec basiert.
- ADVPN ist inkompatibel mit Cisco DMVPN, das auf mGRE-over-IPsec und NHRP basiert.
- **IKE:**
  - IKEv1 : Der **Main-Mode** wird unterstützt (Pre-Shared Key & Certificate Authentication).
  - IKEv1 : Der **aggressive Modus** wird ab FortiOS6.0.1 (Pre-Shared Key & Certificate Authentication) unterstützt.
  - IKEv2 : wird ab FortiOS 5.6.1 unterstützt.
- **Dynamisches Routing:**
  - **BGP und RIPv2/RIPngare** ist unterstützt
  - **PIM/Multicast** wird ab FortiOS5.6.1 unterstützt.
  - **OSPF und IS-IS** werden nicht unterstützt.

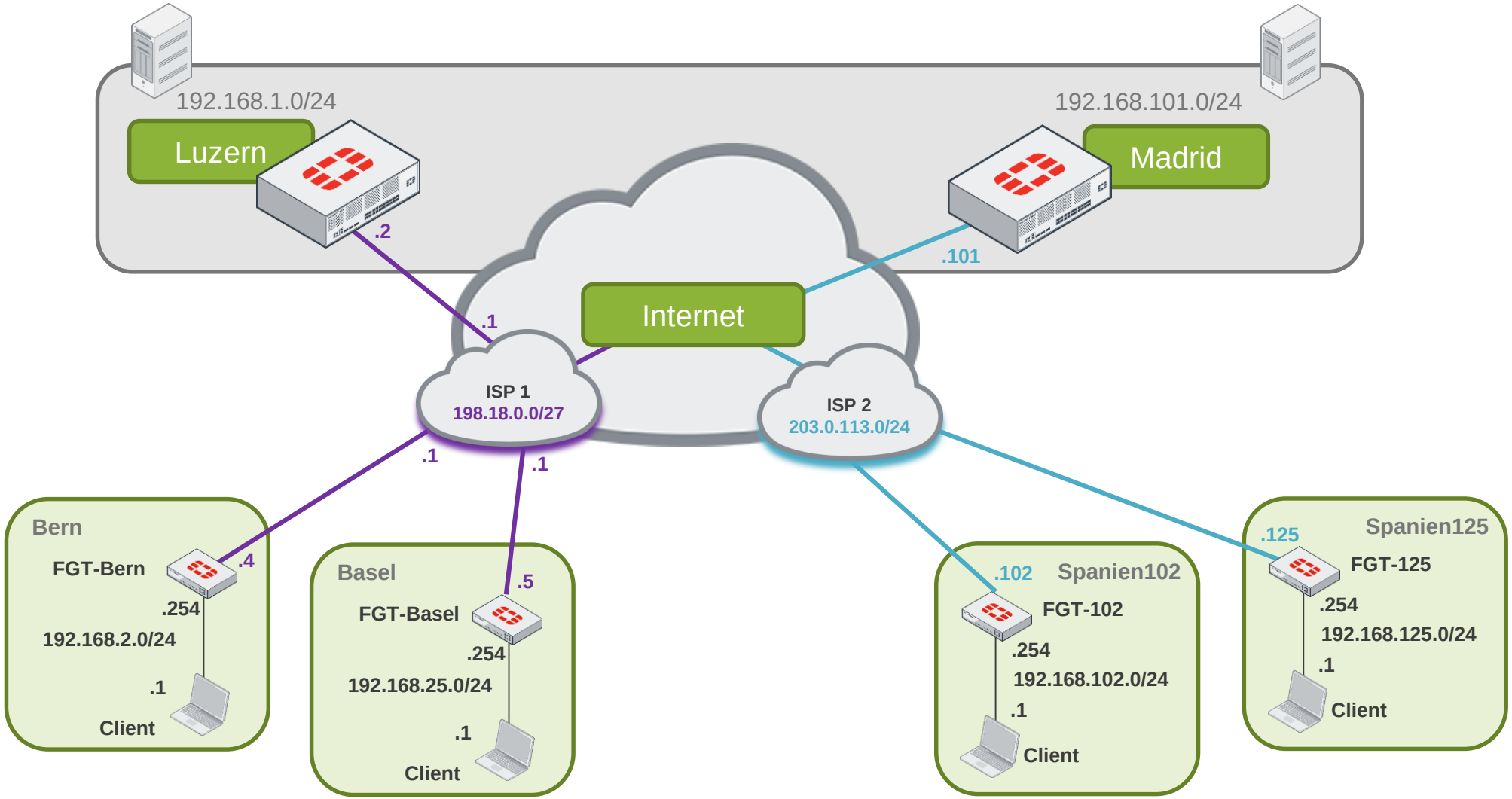
ADVPN Konfiguration und Beispiel

Der Traffic zwischen Bern und Luzern wird die dynamische Aushandlung eines direkten VPN zwischen beiden Standorten auslösen.

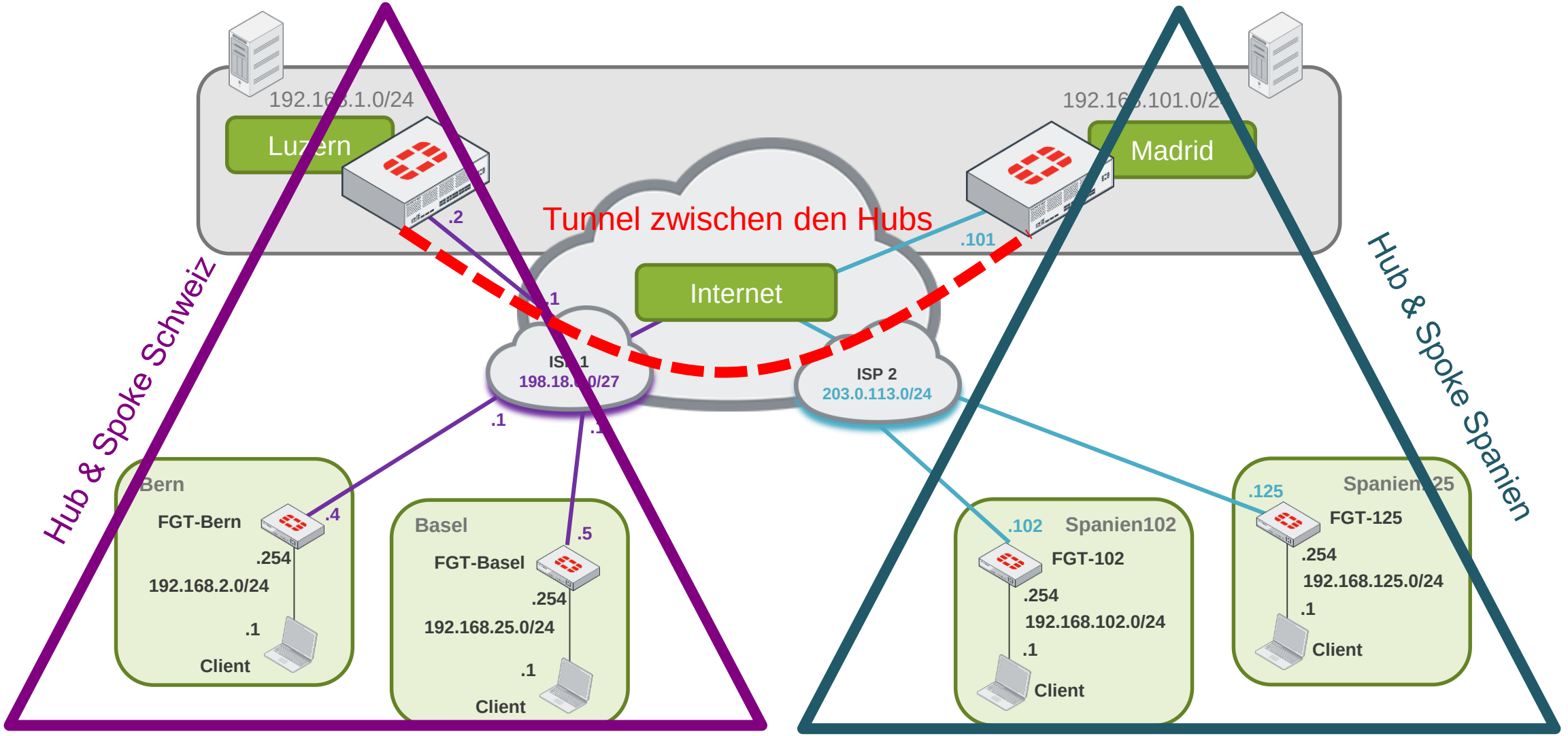


ADVPN Konfiguration

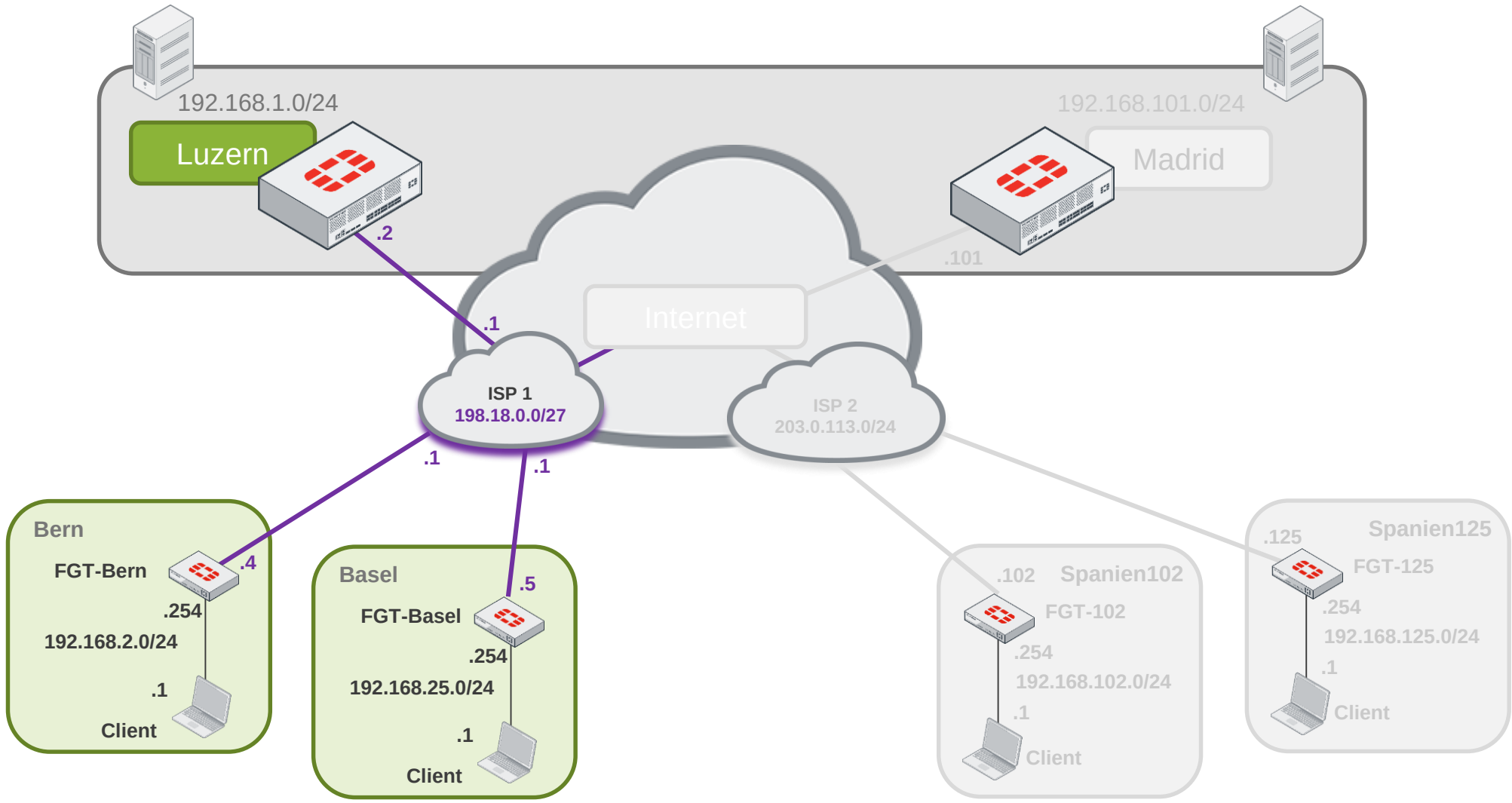
Dual Region Hub & Spoke



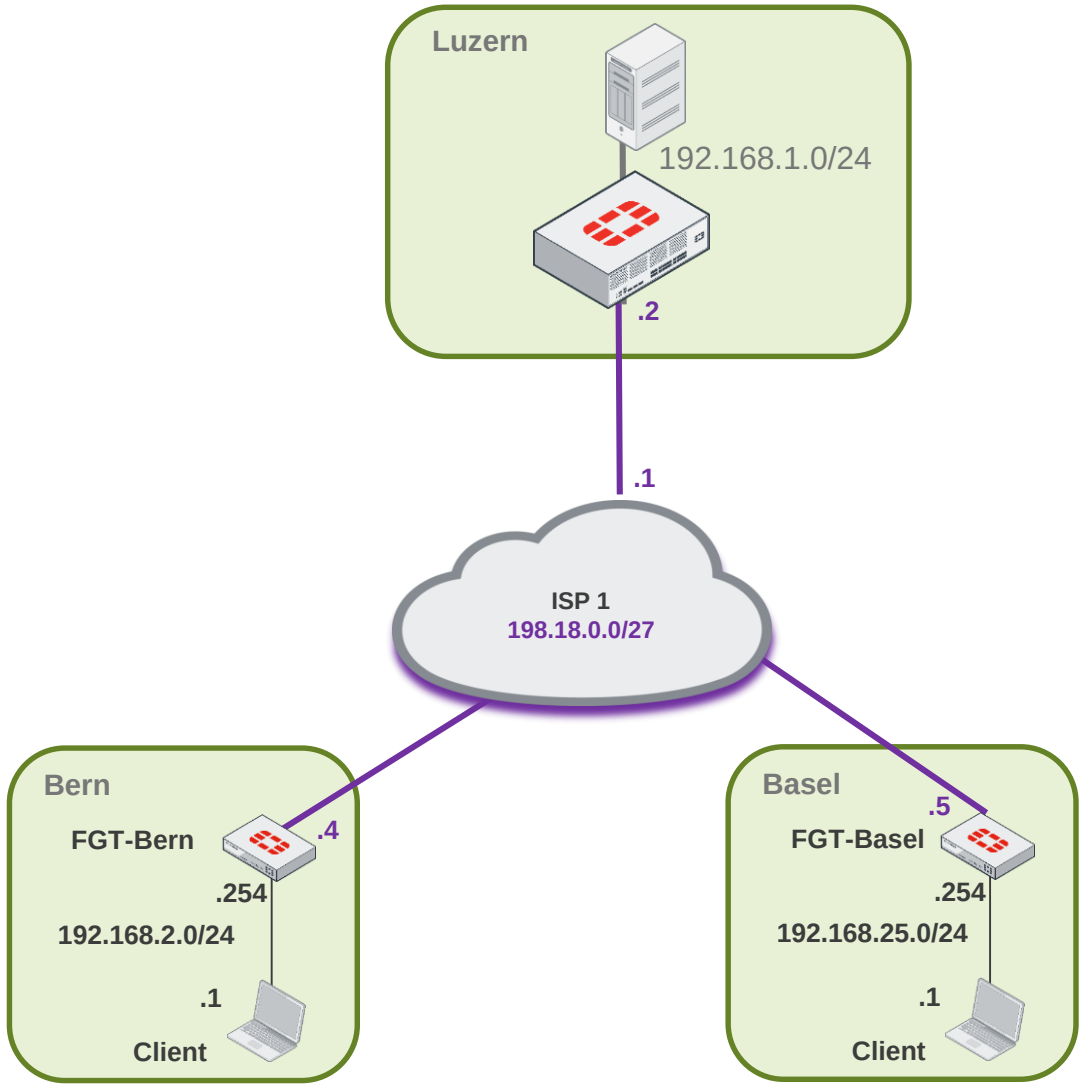
Dual Region Hub & Spoke (Schweiz – Spanien)



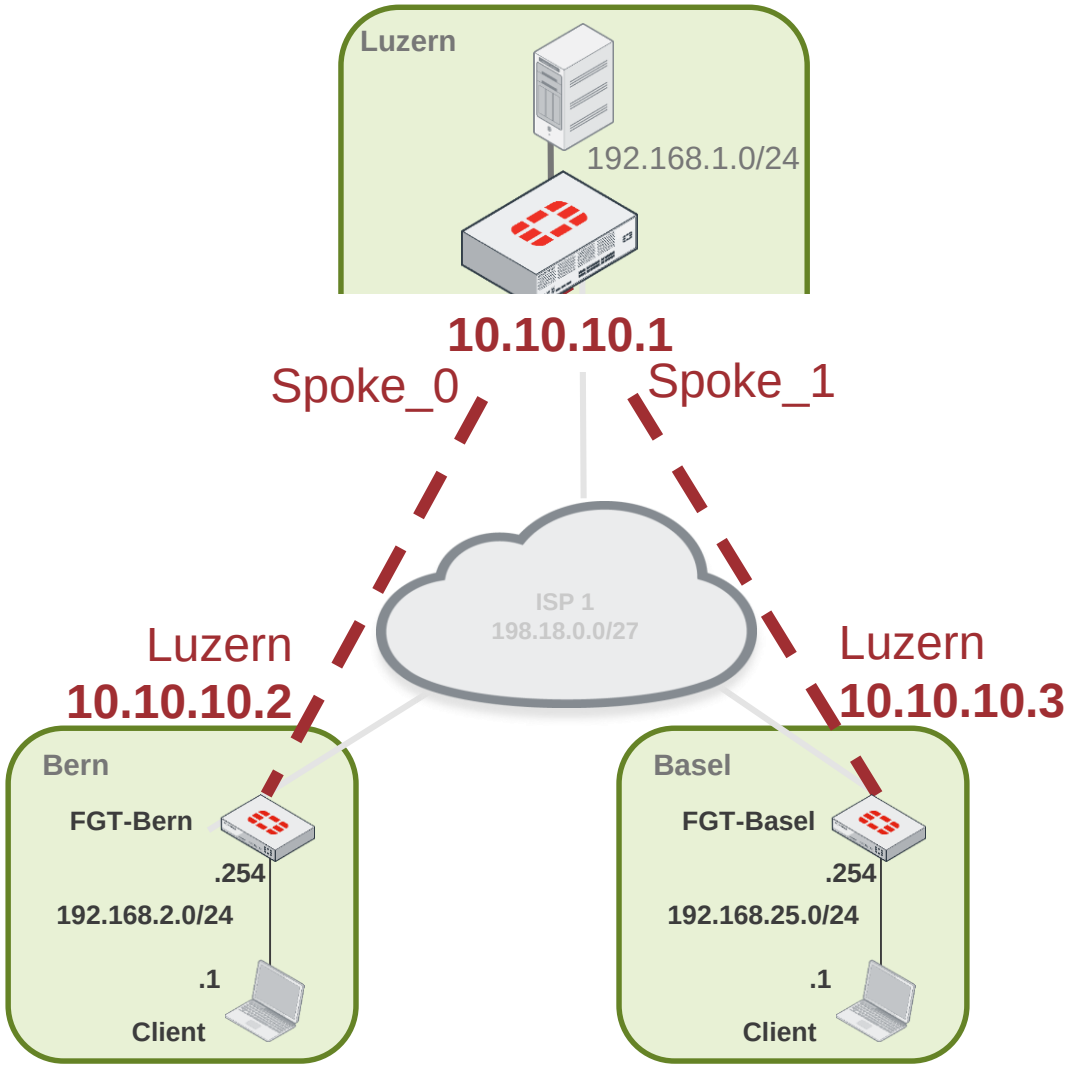
Schweizer Hub & Spoke Netzwerk



Schweizer Hub & Spoke Netzwerk



Schweizer Hub & Spoke Netzwerk **Overlay Subnet**



ADVPN HUB Konfiguration – Parameter Phase 1

- **`auto-discovery-sender enable`**
  Bewirkt, dass der IPsec-Traffic, welcher den Hub passiert, den Hub passieren soll. Es wird ein SHORTCUT Vorschlag(Shortcut-offer) an den Initiator des Traffics gesendet, um darauf hinzuweisen, dass es eventuell eine direktere und bessere Verbindung (SHORTCUT) geben könnte.
- **`add-route disable`**
  Unterbindet das IKE nicht automatisch eine Route zurück über den Spoke hinzufügt, sondern das Routing an ein separat konfiguriertes Routing-Protokoll überlässt.

ADVPN HUB Konfiguration – Parameter Phase 1

- **`net-device disable`**
  Ab FortiOS5.6.3 und 6.0 eine Standardeinstellung für die Dialup Phase1. Es wird nicht mehr ein dediziertes Interface pro Dialer erstellt. Stattdessen wird ein gemeinsames Interface verwendet.
- **`tunnel-search nexthop`**
  Es wird die übereinstimmende Route im Paket als Next-Hop-IP verwendet, um zu entscheiden in welchen Tunnel das Paket dann gesendet werden soll.
- Im FortiOS 5.6.3 und 5.6.4 können die `net-device` und `tunnel-search` Einstellungen nicht mehr modifiziert werden, nach dem die Phase 1 erstellt wurde. Im FortiOS 5.6.5 und 6.0 ist diese Limitation nicht mehr vorhanden

ADVPN Hub Konfiguration – Phase 1 und Phase 2

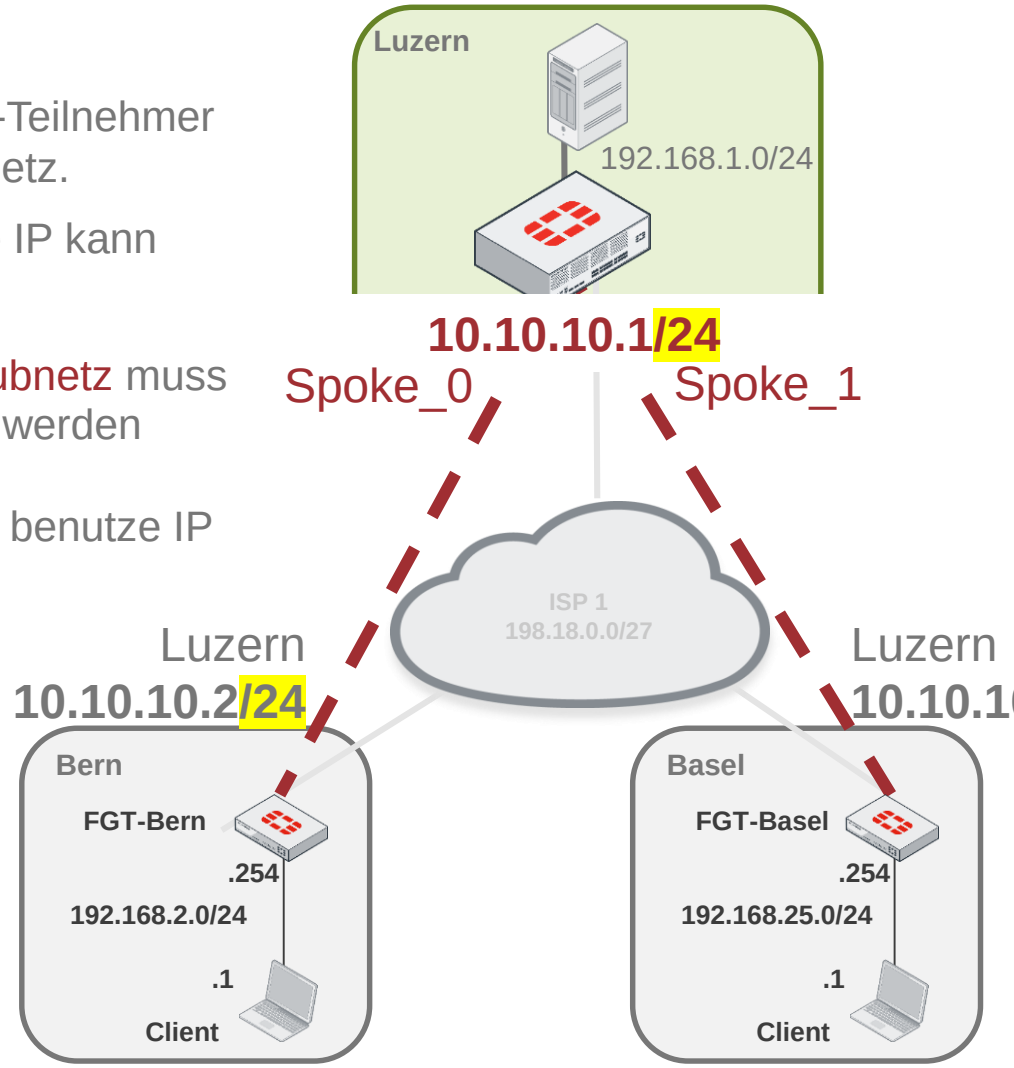
/24

Die Overlay-IPs aller ADVPN-Teilnehmer befinden sich im selben Subnetz.

Die Netzmaske für die Lokale IP kann nur /32 sein.

Die Maske für das Overlay Subnetz muss in «`remote-ip`» angegeben werden

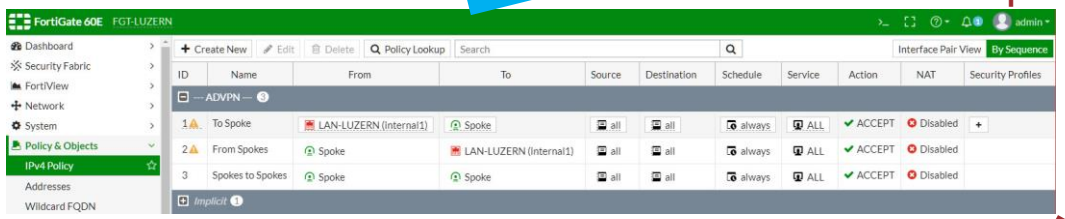
Die `remote-ip` ist eine nicht benutze IP aus dem overlay Subnetz

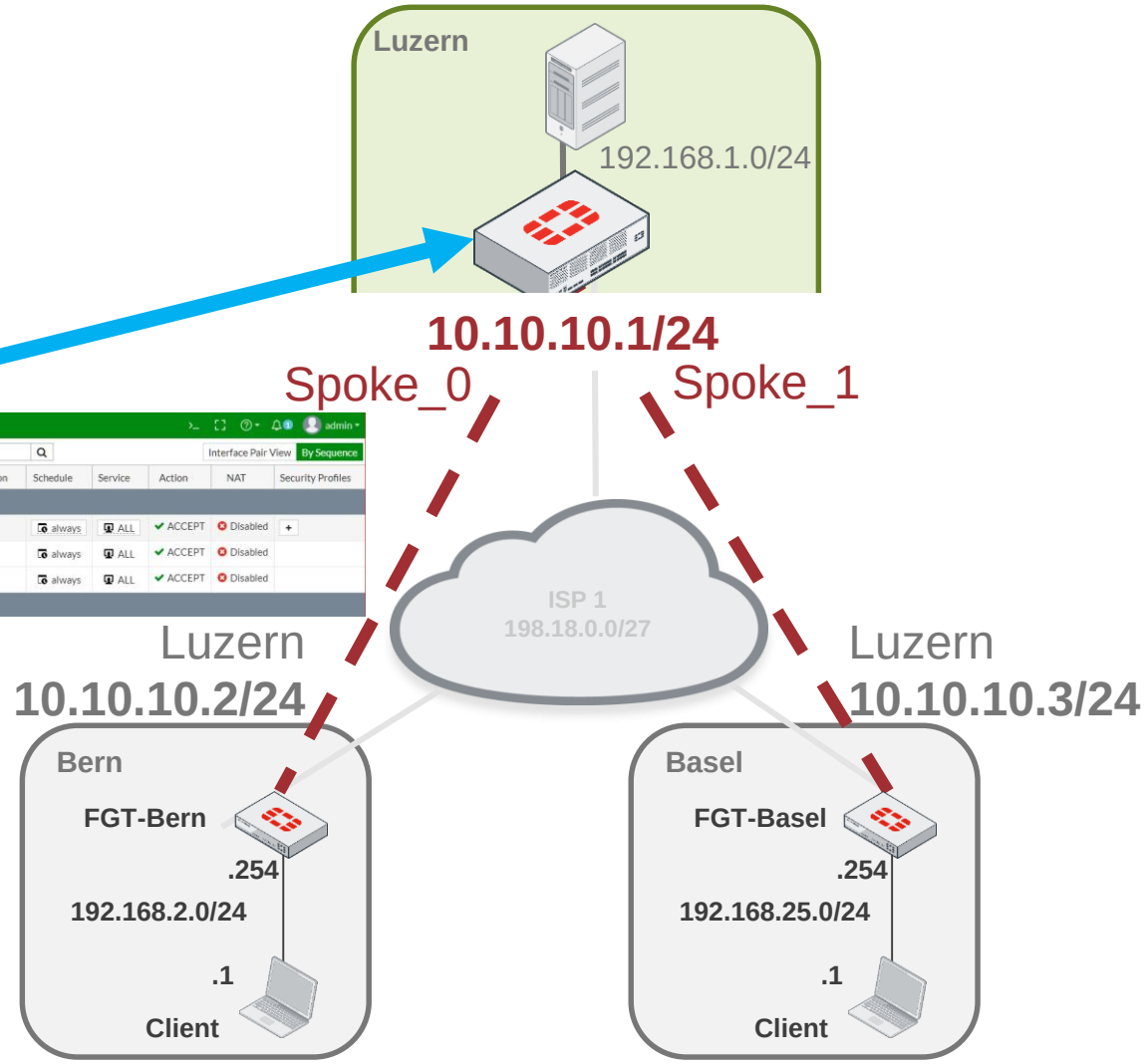



```
config vpn ipsec phase1-interface
    edit "Spoke"
        set type dynamic
        set net-device disable
        set tunnel-search nexthop
        set interface "wan1"
        set proposal aes128-sha1
        set auto-discovery-sender enable
        set add-route disable
        set psksecret fortinet
    next
end

config vpn ipsec phase2-interface
    edit "Spoke"
        set phase1name "Spoke"
        set proposal aes128-sha1
    next
end

config system interface
    edit "Spoke"
        set ip 10.10.10.1/32
        set remote-ip 10.10.10.254/24
    next
end
```

ADVPN Hub Konfiguration - Policies

```
config firewall policy
     edit 1
         set name "To Spokes"
         set srcintf"internalq"
         set dstintf"Spoke"
         set srcaddr"all"
         set dstaddr"all"
         set action accept
         set schedule "always"
         set service "ALL"
     next
     edit 2
         set name "From Spokes"
         set srcintf"Spoke"
         set dstintf"internalq"
         set srcaddr"all"
         set dstaddr"all"
         set action accept
         set schedule "always"
         set service "ALL"
     next
     edit 3
         set name "Spokes to Spokes"
         set srcintf"Spoke"
         set dstintf"Spoke"
         set srcaddr"all"
         set dstaddr"all"
         set action accept
         set schedule "always"
         set service "ALL"
     next
end
```

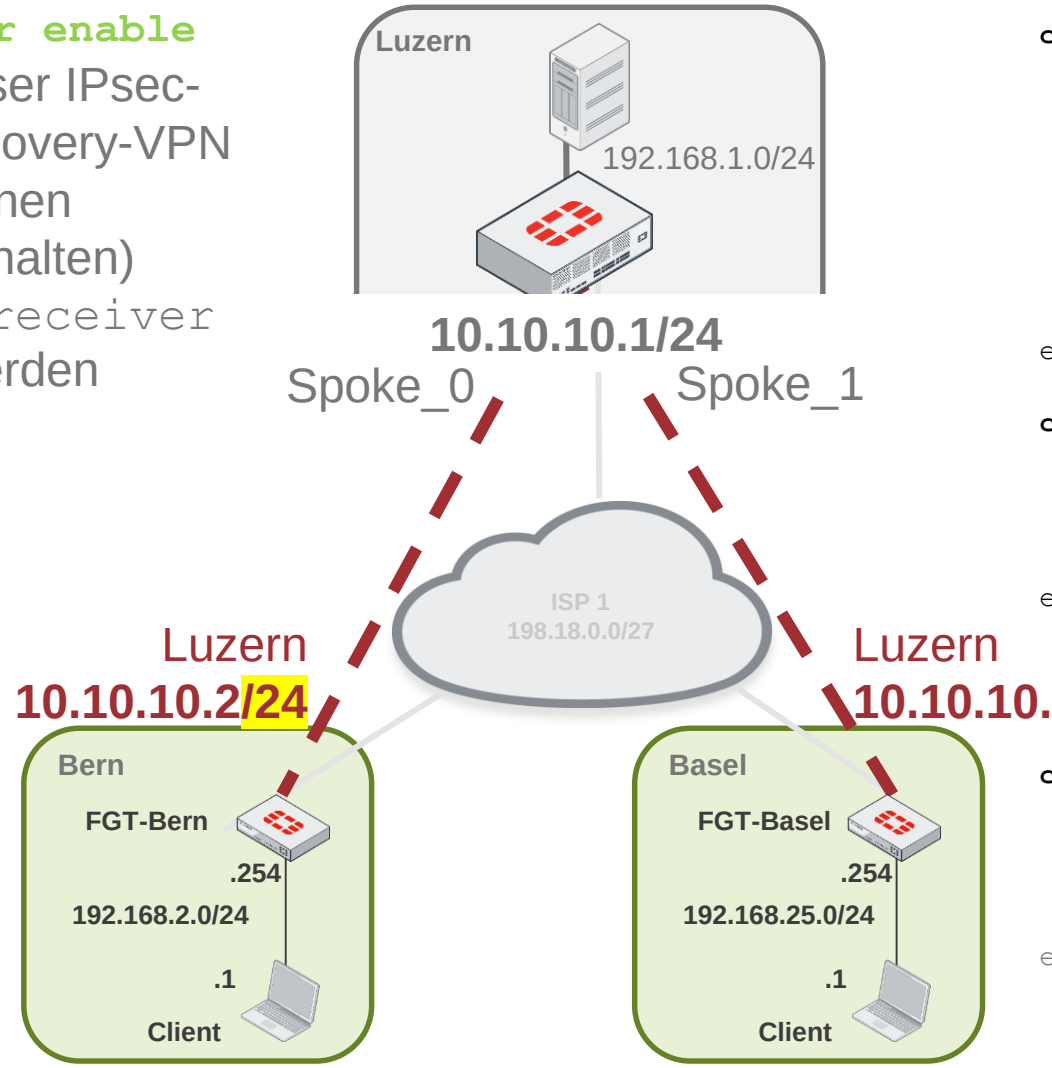




ADVPN Spoke Konfiguration – Phase 1 und Phase 2

`auto-discovery-receiver enable`

Um zu markieren dass dieser IPsec-Tunnel an einem Auto-Discovery-VPN teilnehmen möchte (d.h. einen SHORTCUT-OFFER zu erhalten) muss `auto-discovery-receiver` auf **`enable`** konfiguriert werden




```
config vpn ipsec phase1-interface
    edit "Luzern"
        set interface "wan1"
        set proposal aes128-sha1
        set auto-discovery-receiver enable
        set add-route disable
        set remote-gw 198.18.0.2
        set psksecret fortinet
    next
end

config vpn ipsec phase2-interface
    edit "Luzern"
        set phase1name "Luzern"
        set proposal aes128-sha1
    next
end
```

/24

Die Overlay-IPs aller ADVPN-Teilnehmer befinden sich im selben Subnetz.

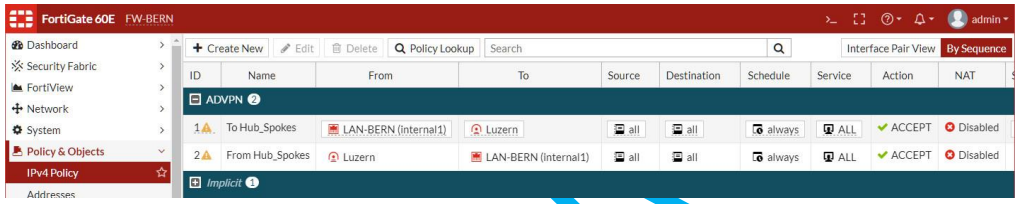
Die Maske für das Overlay Subnetz muss in «`remote-ip`» angegeben werden

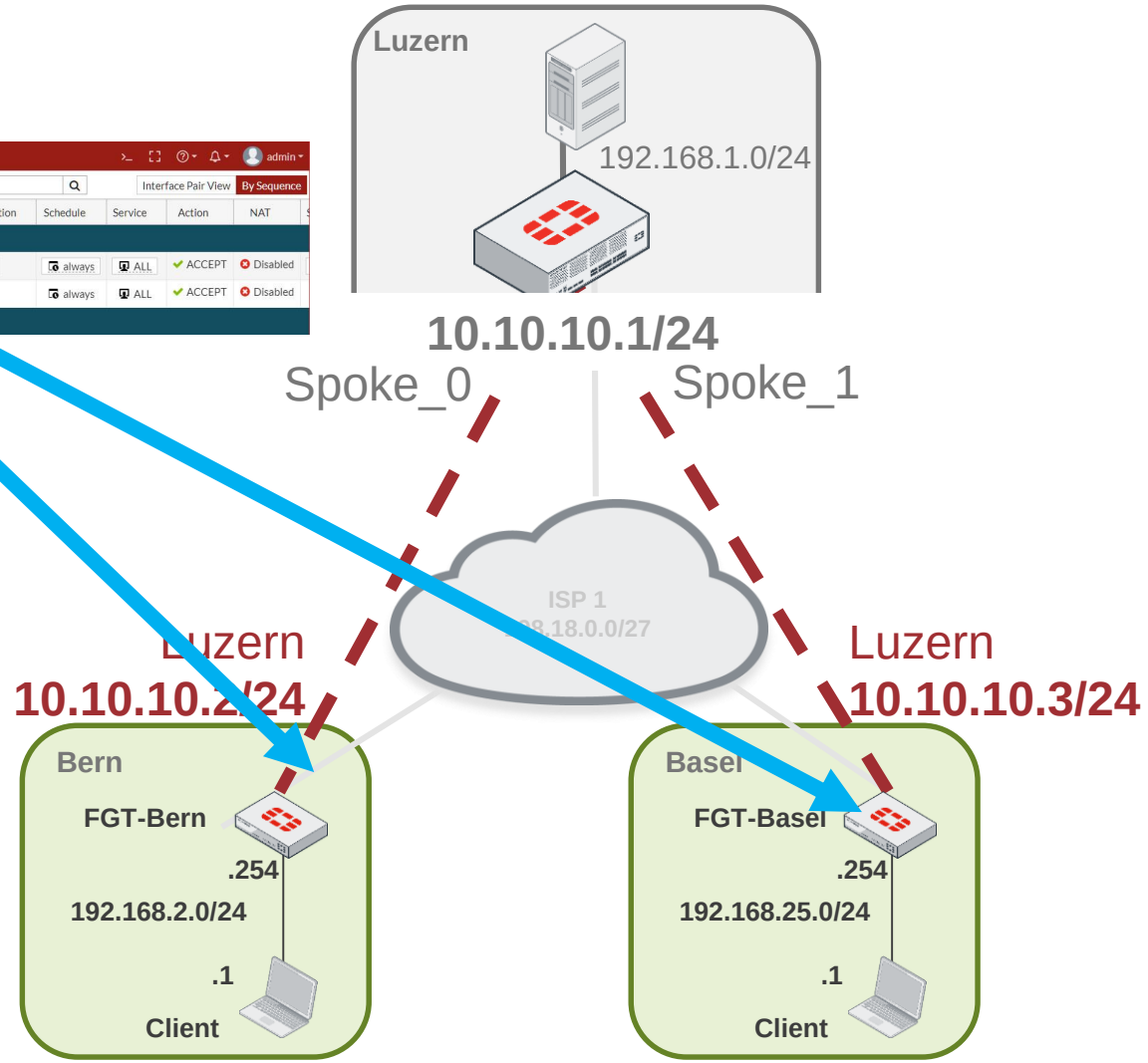
Die `remote-ip` kann jede andere IP im Overlay sein, aus Gründen der Übersichtlichkeit wird die IP des Hubs verwendet.

```
config system interface
    edit "Luzern"
        set ip 10.10.10.2/32 (Basel .3)
        set remote-ip 10.10.10.1/24
    next
end
```

ADVPN Spoke Konfiguration - Policies

```
config firewall policy
     edit 1
         set name "To Hub_Spokes"
         set srcintf "internal1"
         set dstintf "Luzern"
         set srcaddr "all"
         set dstaddr "all"
         set action accept
         set schedule "always"
         set service "ALL"
     next
     edit 2
         set name "From Hub_Spokes"
         set srcintf "Luzern"
         set dstintf "internal1"
         set srcaddr "all"
         set dstaddr "all"
         set action accept
         set schedule "always"
         set service "ALL"
     next
End
```

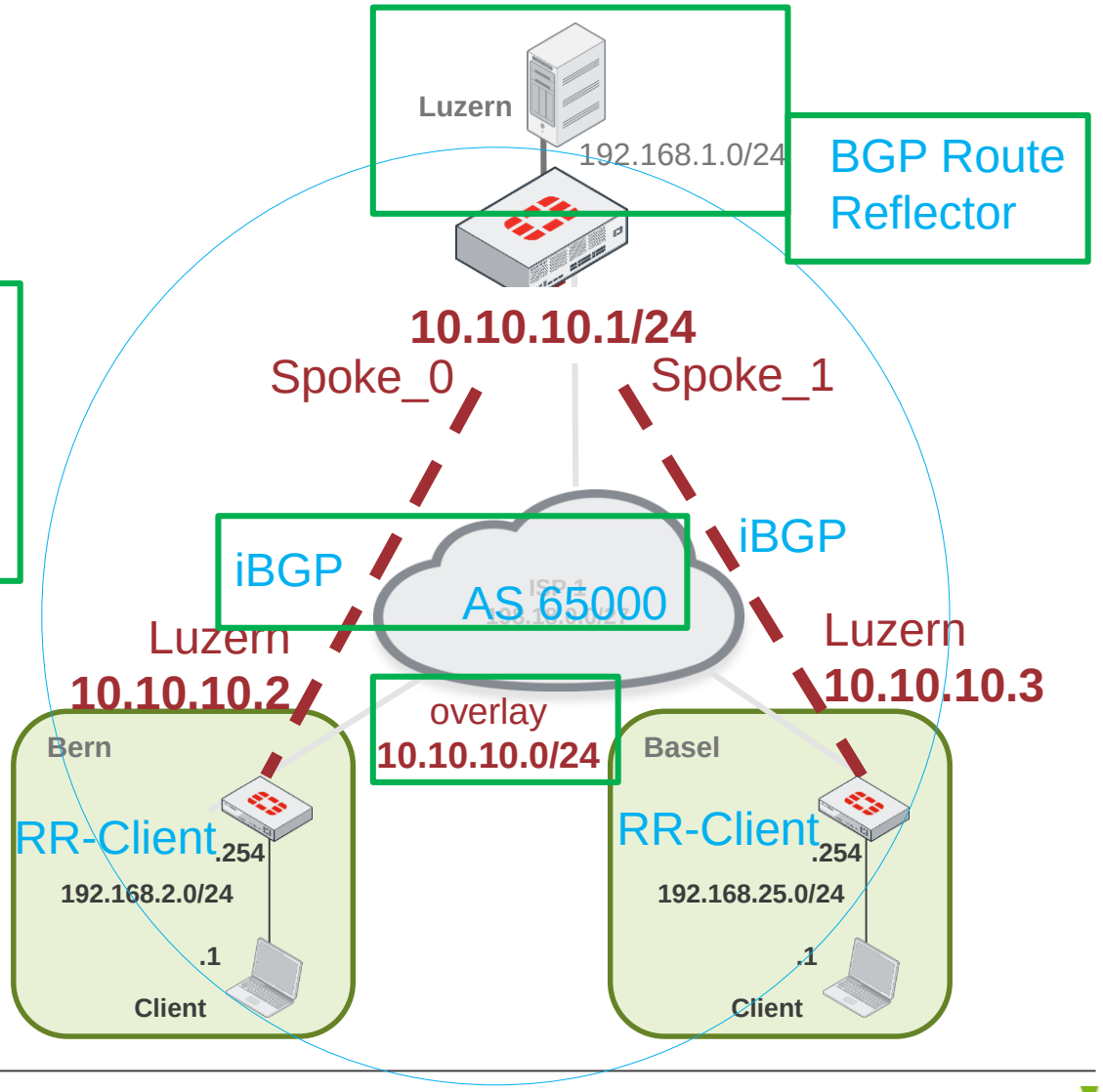




- Für den Traffic von/zu den anderen Spokes sind keine spezifischen Regeln erforderlich.
- Der Traffic von/zu anderen Spokes wird anhand der Policies vom/zum Hub überprüft.

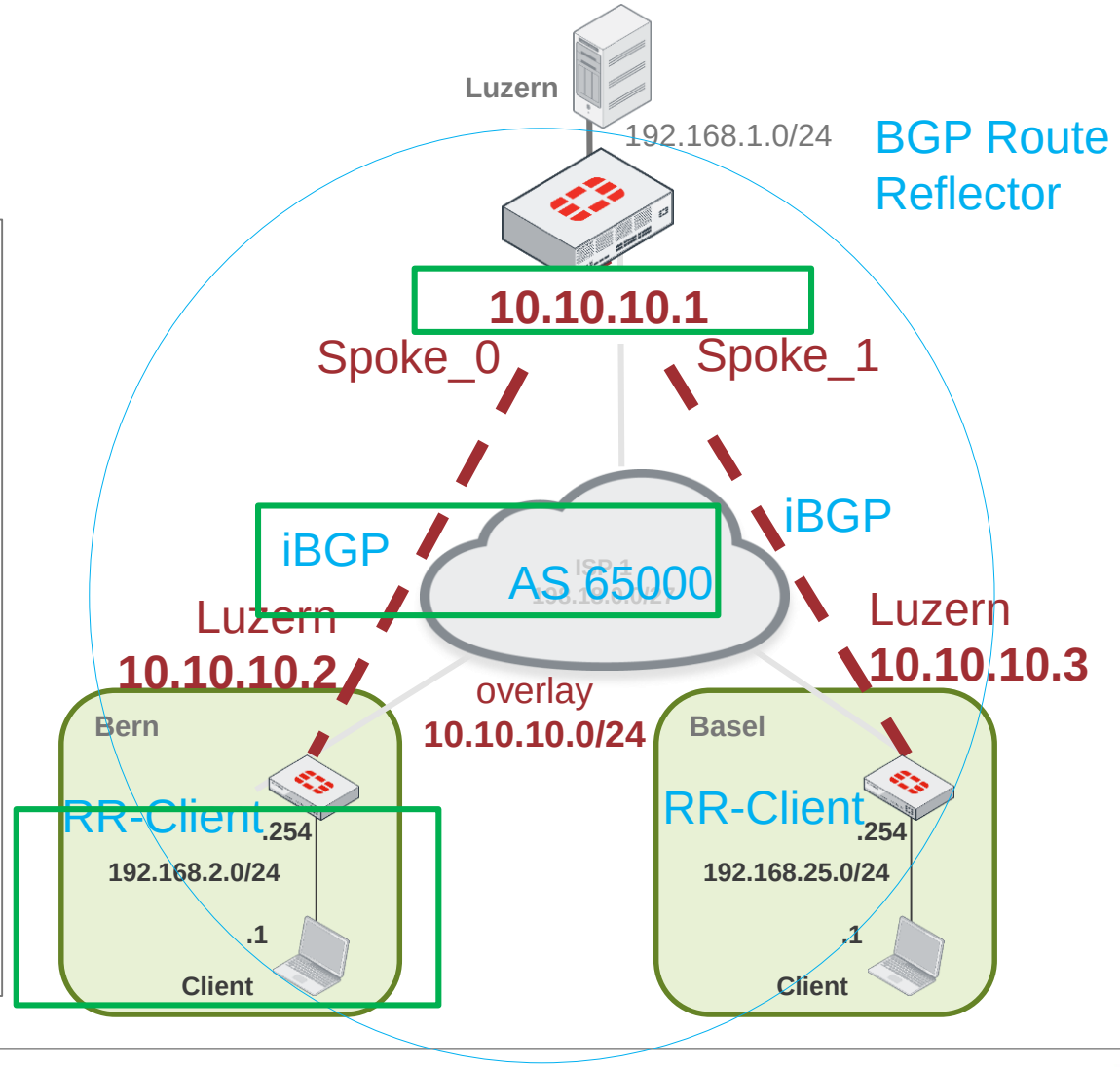
Hub Konfiguration = iBGP Route Reflector (RR)

```
config router bgp
     set as 65000
     set router-id 10.10.10.1
     config neighbor-group
          edit «advn_peers»
               set remote-as 65000
               set route-reflector-client enable
          next
     end
     config neighbor-range
          edit 1
               set prefix 10.10.10.0 255.255.255.0
               set neighbor-group «advn_peers»
          next
     end
     config network
          edit 1
               set prefix 192.168.1.0 255.255.255.0
          next
     end
end
```



Spoke Konfiguration = iBGP RR-Client

```
config router bgp
    set as 65000
    set router-id 10.10.10.2
    config neighbor
        edit «10.10.10.1»
            set remote-as 65000
        next
    end
    config network
        edit 1
            set prefix 192.168.2.0 255.255.255.0
        next
    end
end
```

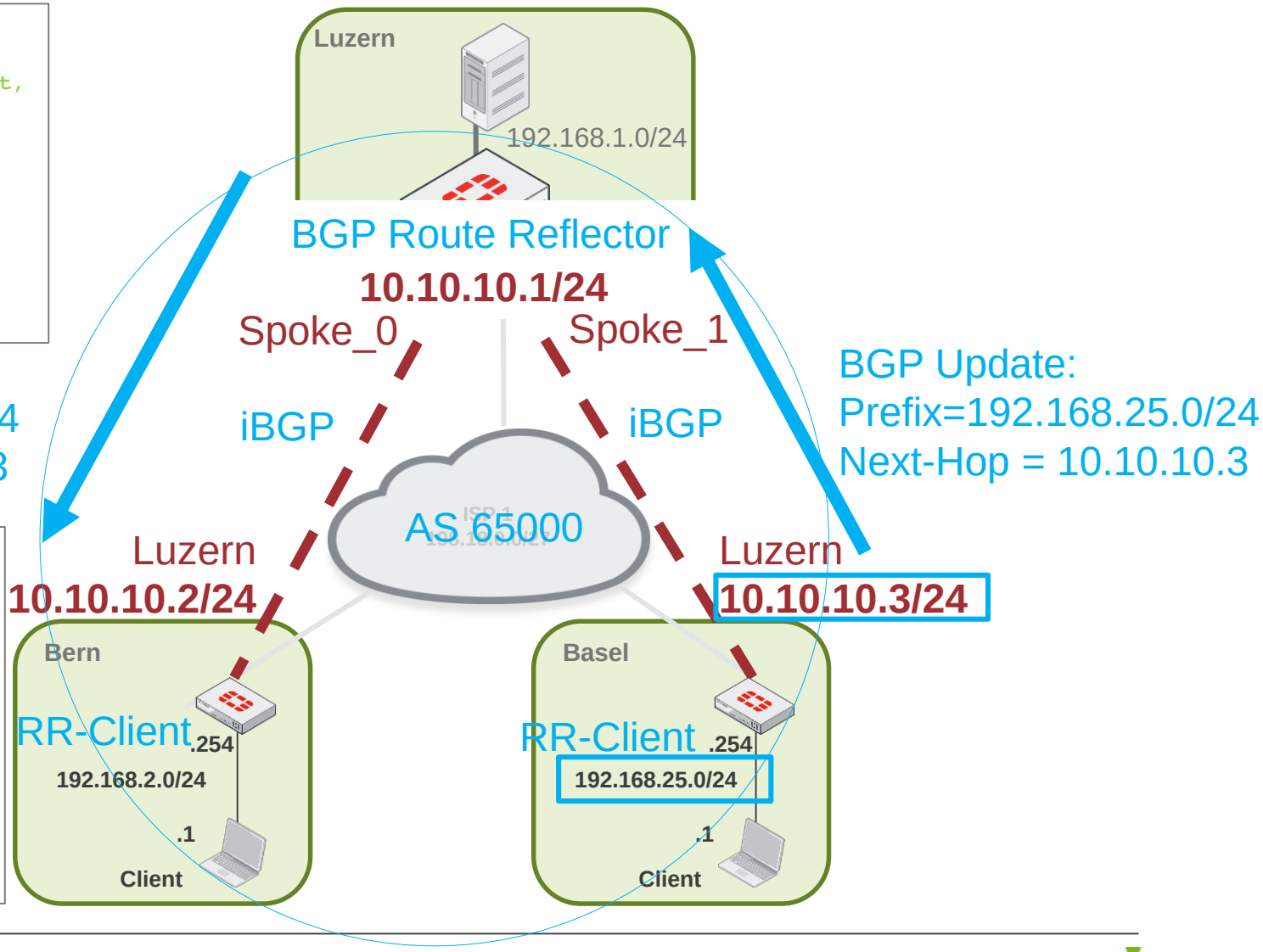


iBGP – Route Reflector (RR) und RR-Clients

```
FGT-LUZERN # get router info bgp network
BGP table version is 4, local router ID is 10.10.10.1
Status codes: s suppressed, d damped, h history, * valid, > best,
I - internal,
              S Stale
Orgin codes:   i - IPG, e - EGP, ? - incomplete

   Netzwork     Next Hop         Metric LocPrf Weight Path
*> 192.168.1.0   0.0.0.0                       100 32768    i
*>i192.168.25.0     10.10.10.3           0        100     0    i
```

BGP Update:
Prefix=192.168.25.0/24
Next-Hop = 10.10.10.3

```
FGT-BERN # get router info bgp network
BGP table version is 4, local router ID 10.10.10.2
Status codes: s suppressed, d damped, h history, * valid, > best ,
i-internal,
              S Stale
Origin codes:    i-IGP, e -EGP, ? -incomplete
   Network     Next Hop         Metric LocPrf Weight Path
*>i192.168.1.0   10.10.10.1        0    100      0     i
*> 192.168.2.0   0.0.0.0                100 32768     i
*>i192.168.25.0  10.10.10.3         0    100      0     i

Total number of prefixes 3
```



BGP Update:
Prefix=192.168.25.0/24
Next-Hop = 10.10.10.3

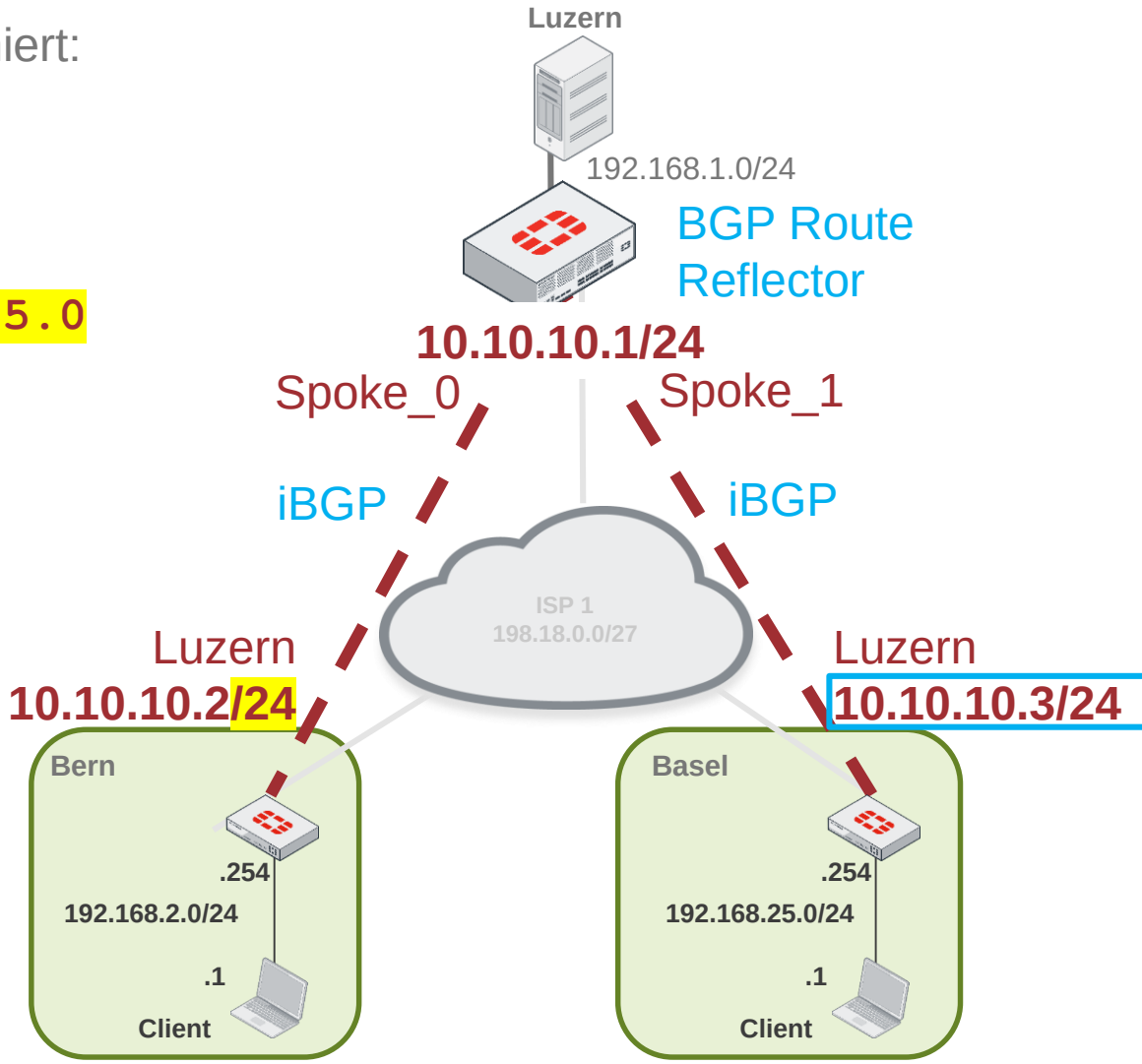
iBGP – Next Hop Reachability

Das ADVPN-Overlay-Subnetz wird am Tunnel Interface definiert:

```
config system interface
      edit «Luzern»
            set ip 10.10.10.2 255.255.255.255
            set remote-ip 10.10.10.1 255.255.255.0
      next
end
```

BGP Next-Hop muss durch den Tunnel zugänglich sein.

```
FGT-BERN # get router info bgp network
BGP table version is 4, local router ID 10.10.10.2
Status codes: s suppressed, d damped, h history, * valid, > best ,
i-internal,
              S Stale
Origin codes:              i-IGP, e -EGP, ? -incomplete
  Network  Next Hop       Metric LocPrf Weight Path
*>i192.168.1.0            10.10.10.1    0    100      0    i
*> 192.168.2.0            0.0.0.0                100 32768    i
*>i192.168.25.0           10.10.10.3               0    100      0
i

Total number of prefixes 3
```



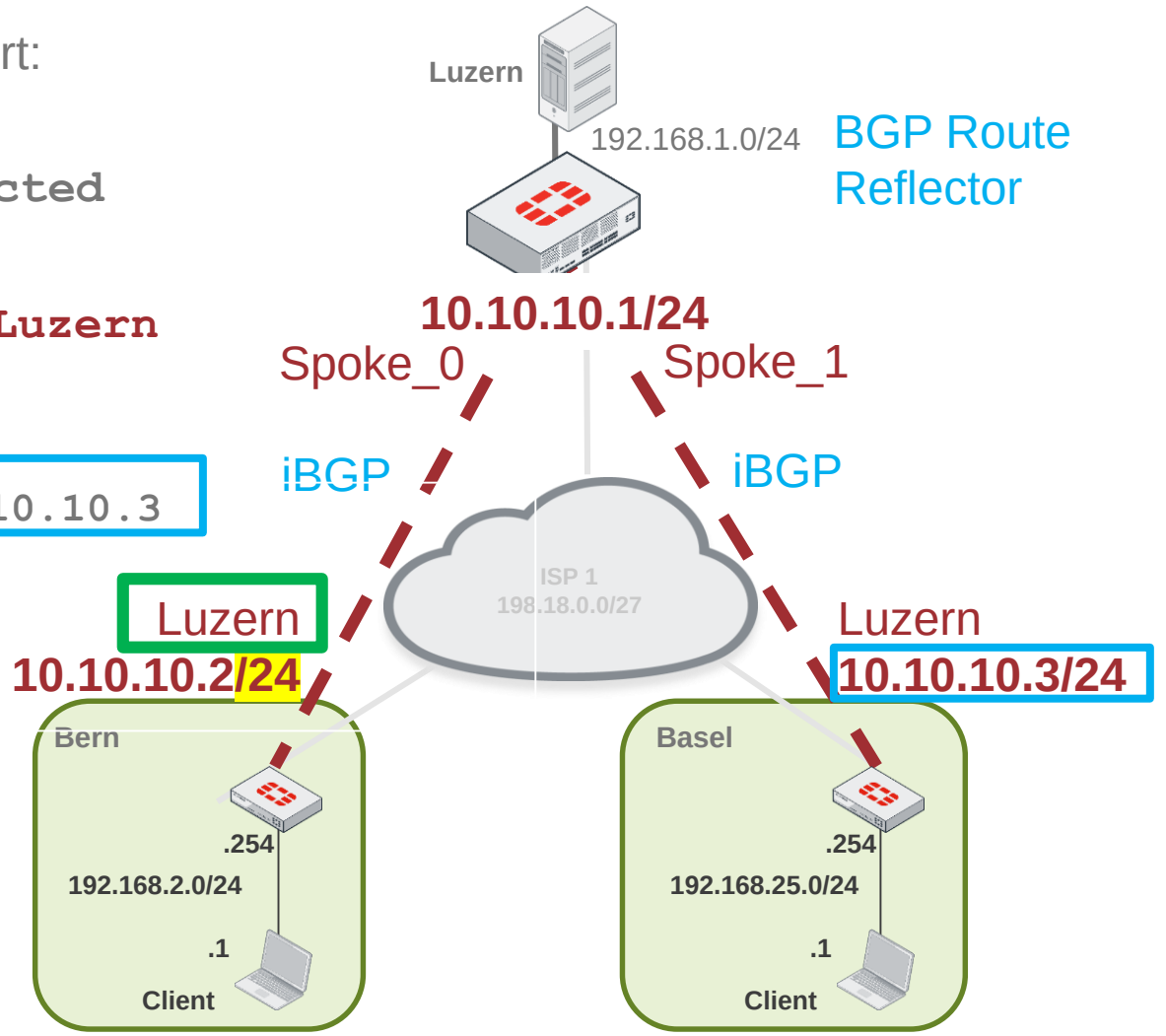
Kein SHORTCUT – BGP Next-Hop ist erreichbar via HUB

Das ADVPN-Overlay-Subnetz wird am Tunnel Interface definiert:

```
FGT-BERN # get router info routing-table connected
(...)
C        10.10.10.0/24 is directly connected, Luzern
```

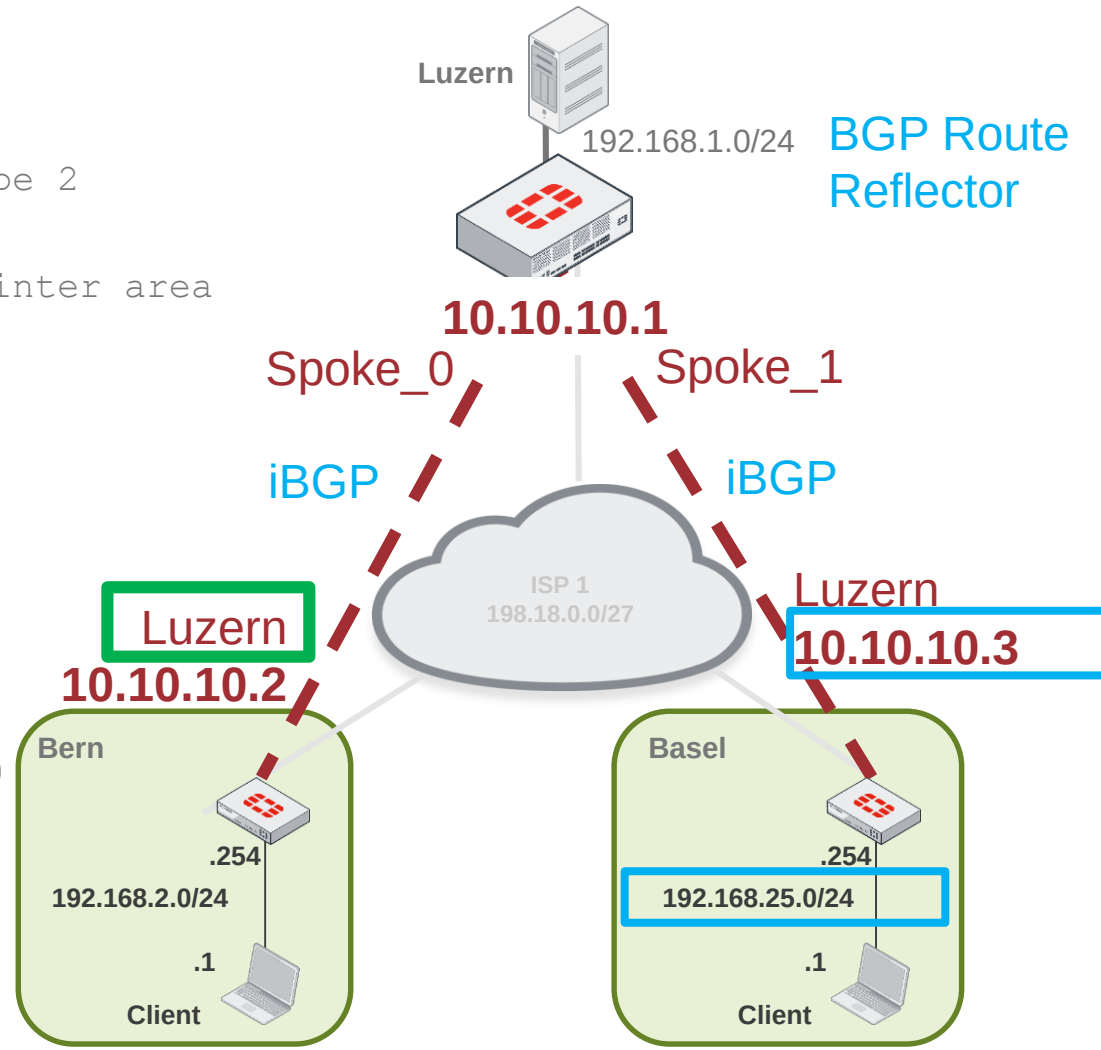
```
FGT-BERN # get router info routing-table details 10.10.10.3
Routing entry for 10.10.10.0/24
  Known via "connected", distance 0, metric 0, best
  * is directly connected, Luzern
```

BGP Next-Hop für Spoke Basel (**10.10.10.10.3**) ist über das angeschlossene Subnetz von **Luzern** erreichbar.



Kein SHORTCUT – RIB lookup

```
FGT-BERN # get router info routing-table all
Codes:  K -kernel, C -connected, S -static, R -RIP, B -BGP
        O -OSPF, IA -OSPF inter area
        N1 -OSPF NSSA external type 1, N2 -OSPF NSSA external type 2
        E1 -OSPF external type 1, E2 -OSPF external type 2
        i-IS-IS, L1 -IS-IS level-1, L2 -IS-IS level-2, ia-IS-IS inter area
        * -candidate default
S*       0.0.0.0/0 [10/0] via 198.18.0.1, wan1

C        10.10.10.0/24 is directly connected, Luzern
                       is directly connected, Luzern

B        192.168.1.0/24 [200/0] via 10.10.10.1, Luzern, 03:51:14
C        192.168.2.0/24 is directly connected, internal1
B        192.168.25.0/24 [200/0] via 10.10.10.3, Luzern, 00:16:20
B        192.168.4.0/24 [200/0] via 10.10.10.4, Luzern, 00:16:20
B        192.168.5.0/24 [200/0] via 10.10.10.5, Luzern, 00:16:20
C        198.18.0.0/27  is directly connected, wan1
```

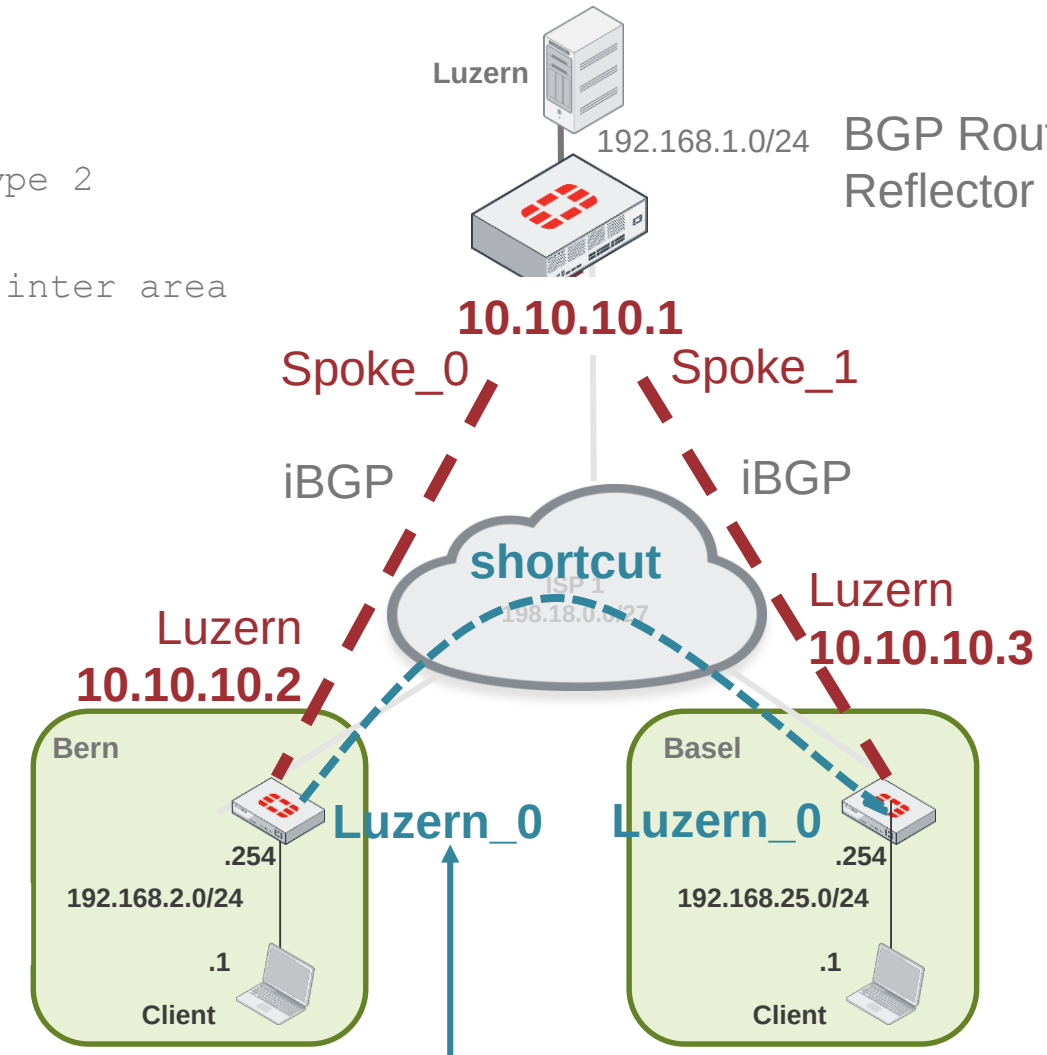


Der Spoke zu Spoke Traffic geht über den Hub in Luzern

Mit SHORTCUT – ADVPN Overlay

```
FGT-BERN # get router info routing-table all
Codes:  K -kernel, C -connected, S -static, R -RIP, B -BGP
        O -OSPF, IA -OSPF inter area
        N1 -OSPF NSSA external type 1, N2 -OSPF NSSA external type 2
        E1 -OSPF external type 1, E2 -OSPF external type 2
        i-IS-IS, L1 -IS-IS level-1, L2 -IS-IS level-2, ia-IS-IS inter area
        * -candidate default

S*      0.0.0.0/0 [10/0] via 198.18.0..254, wan1
S       10.10.10.0/24 [10/0] via 10.10.10.1, Luzern
C       10.10.10.1/32 is directly connected, Luzern
C       10.10.10.2/32 is directly connected, Luzern
                      is directly connected, Luzern_0
C       10.10.10.3/32 is directly connected, Luzern_0
B       192.168.1.0/24 [200/0] via 10.10.10.1, Luzern, 02:38:15
C       192.168.2.0/24 is directly connected, internal1
B       192.168.25.0/24 [200/0] via 10.10.10.3, Luzern_0, 00:00:28
C       198.18.0..0/24 is directly connected, wan1
```

Shortcut Name = <phase1name>_<index>

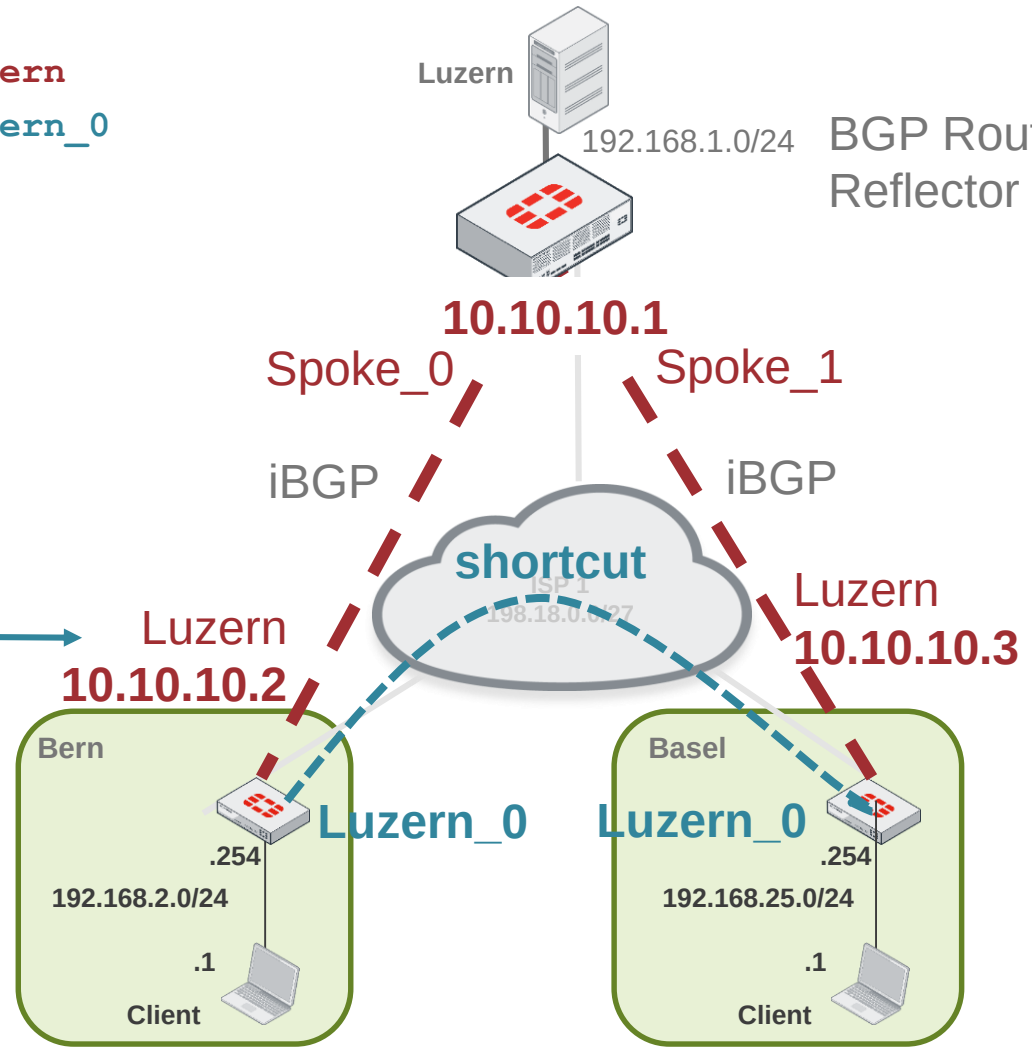


Mit SHORTCUT – ADVPN Overlay

```
FGT-BERN # diag ip address list | grep Luzern
IP=10.10.10.2 -> 10.10.10.1/255.255.255.255 index=15 devname=Luzern
IP=10.10.10.2 -> 10.10.10.3/255.255.255.255 index=19 devname=Luzern_0
```

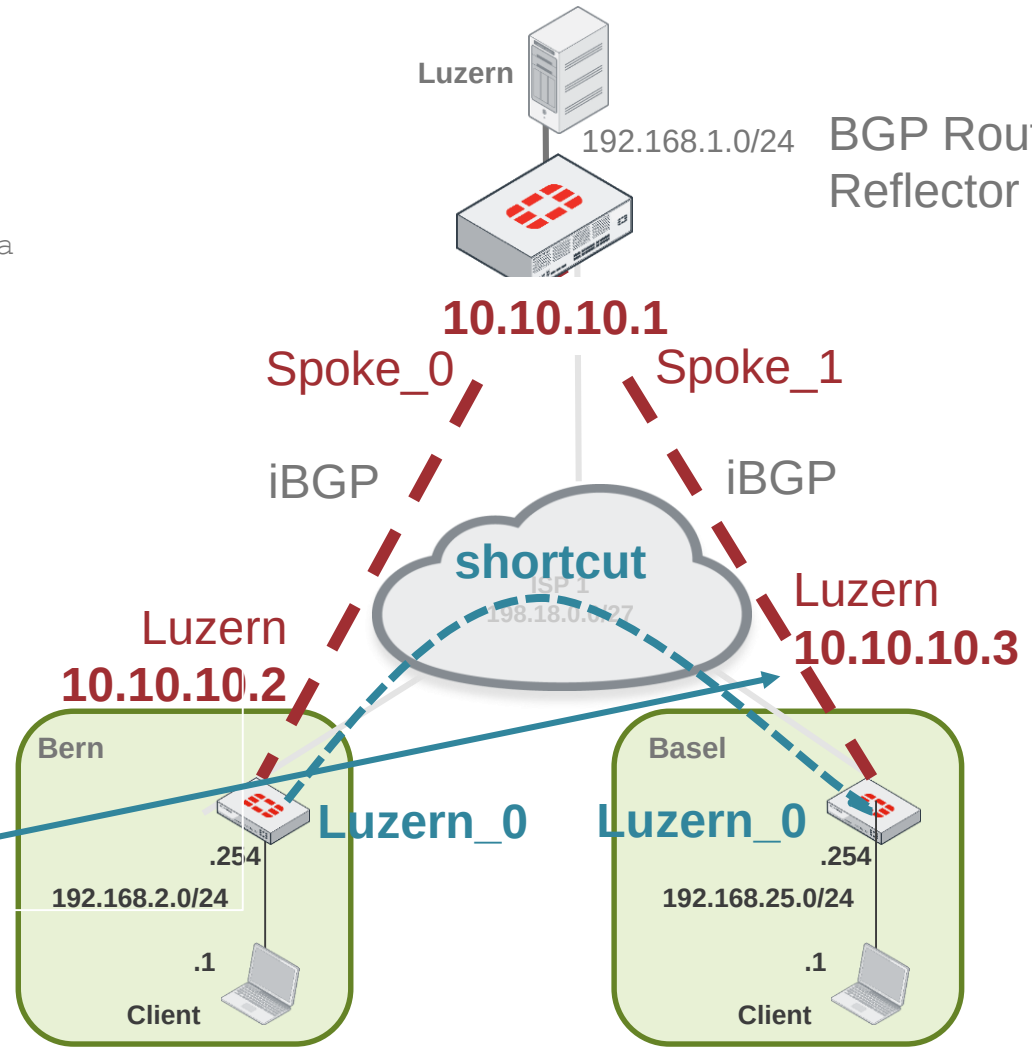
Die gleiche Overlay-IP wird zugewiesen:

- der Tunnel zum Hub
- die Shortcuts zu anderen Spoke(s)



Mit SHORTCUT – BGP Next-Hop ist direkt verbunden

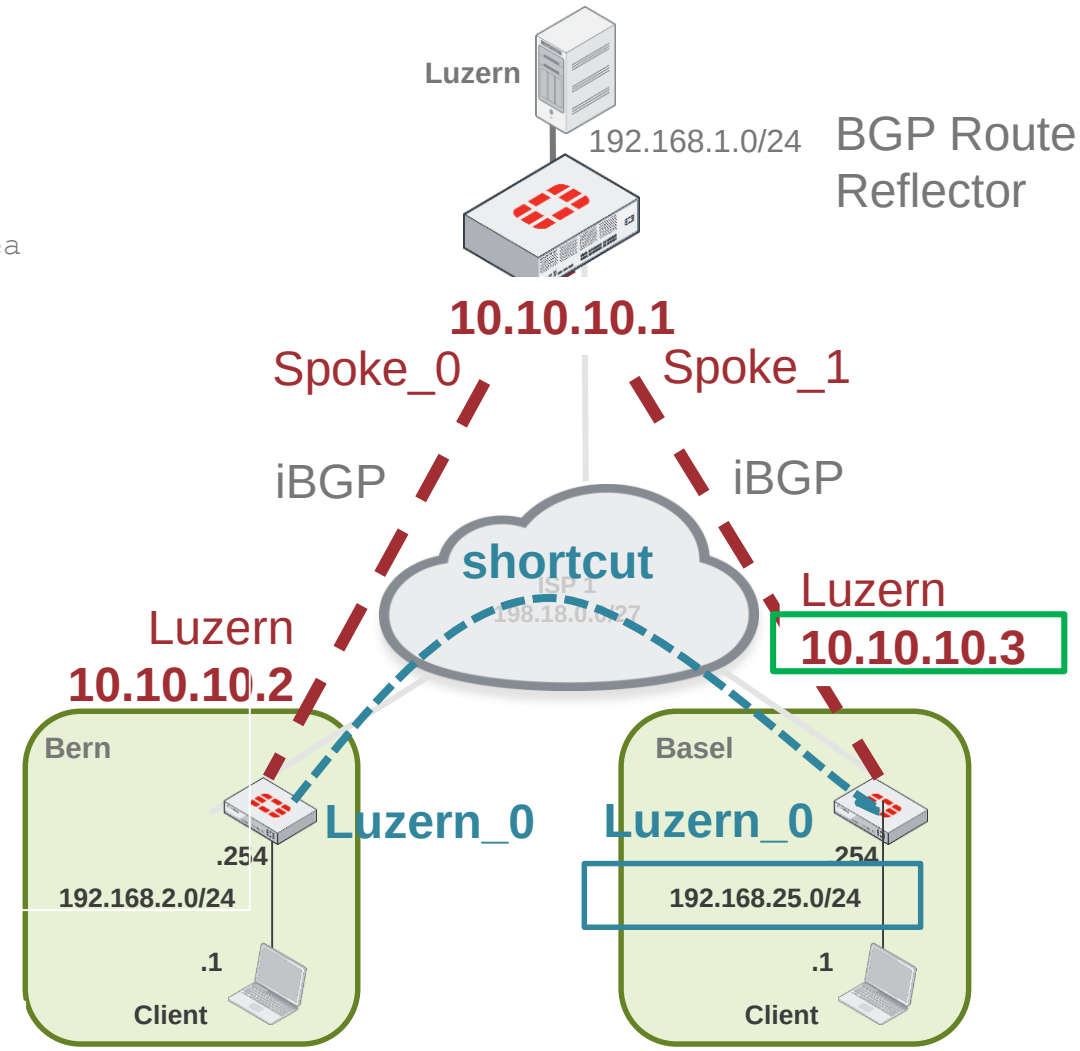
```
FGT-BERN # get router info routing-table all
Codes:   K -kernel, C -connected, S -static, R -RIP, B -BGP
         O -OSPF, IA -OSPF inter area
         N1 -OSPF NSSA external type 1, N2 -OSPF NSSA external type 2
         E1 -OSPF external type 1, E2 -OSPF external type 2
         i-IS-IS, L1 -IS-IS level-1, L2 -IS-IS level-2, ia-IS-IS inter area
         * -candidate default
S*       0.0.0.0/0 [10/0] via 198.18.0.1, wan1

S        10.10.10.0/24 [10/0] via 10.10.10.1, Luzern

C        10.10.10.1/32 is directly connected, Luzern
C        10.10.10.2/32 is directly connected, Luzern
                       is directly connected, Luzern_0
C        10.10.10.3/32 is directly connected, Luzern_0
B        192.168.1.0/24 [200/0] via 10.10.10.1, Luzern, 02:38:15
C        192.168.2.0/24 is directly connected, internal1
B        192.168.25.0/24 [200/0] via 10.10.10.3, Luzern_0, 00:00:28

C        198.18.0.1/27 is directly connected, wan1
```



Der BGP Next-Hop von FGT-BASEL ist direkt über das Shortcut-Interface verbunden.

Mit SHORTCUT – RIB lookup

```
FGT-BERN # get router info routing-table all
Codes:   K -kernel, C -connected, S -static, R -RIP, B -BGP
         O -OSPF, IA -OSPF inter area
         N1 -OSPF NSSA external type 1, N2 -OSPF NSSA external type 2
         E1 -OSPF external type 1, E2 -OSPF external type 2
         i-IS-IS, L1 -IS-IS level-1, L2 -IS-IS level-2, ia-IS-IS inter area
         * -candidate default
S*       0.0.0.0/0 [10/0] via 198.18.0.1, wan1

S        10.10.10.0/24 [10/0] via 10.10.10.1, Luzern

C        10.10.10.1/32 is directly connected, Luzern
C        10.10.10.2/32 is directly connected, Luzern
                       is directly connected, Luzern_0
C        10.10.10.3/32 is directly connected, Luzern_0
B        192.168.1.0/24 [200/0] via 10.10.10.1, Luzern, 02:38:15
C        192.168.2.0/24 is directly connected, internal1
B        192.168.25.0/24 [200/0] via 10.10.10.3, Luzern_0, 00:00:28

C        198.18.0.1/27 is directly connected, wan1
```



Spoke to Spoke Traffic geht über den Shortcut

Nützliche CLI Kommandos

Nützliche CLI Kommandos - IPsec

```
FGT_BERN # diag vpn tunnel list
list all ipsec tunnel in vd0
------------------------------------------------------
name=Luzern ver=1 serial=1 198.18.0.4:0->198.18.0.2:0
bound_if=4 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/0
proxyid_num=1 child_num=0 refcnt=21 ilast=1 olast=1 auto-discovery=2
stat: rxp=63 txp=58 rxb=8424 txb=3627
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=Luzern proto=0 sa=1 ref=2 serial=1 adr
src: 0:0.0.0.0/0.0.0.0:0
dst: 0:0.0.0.0/0.0.0.0:0
SA: ref=3 options=2e type=00 soft=0 mtu=1438 expire=42941/0B replaywin=2048 seqno=3b esn=0 replaywin_lastseq=00000040
life: type=01 bytes=0/0 timeout=43172/43200
dec: spi=8135a40e esp=aeskey=16 97e4ad16da7bb2a3880b65311da8e50e
ah=sha1 key=20 968a42ff561c7ea5fd4d8a94546c09f4812f0dcf
enc: spi=e77918f5 esp=aeskey=16 0143920846bd2087bb56250e855a4352
ah=sha1 key=20 328d6e7713f2beb8d44ec4944ca868258236c812
dec:pkts/bytes=63/4278, enc:pkts/bytes=58/7488

FGT_BERN # get vpn ipsec tunnel summary
'Luzern' 198.51.100.1:0  selectors(total,up): 1/1  rx(pkt,err): 66/0  tx(pkt,err): 60/0

FGT_BERN # diag vpn ike status detailed
vd: root/0
name: Luzern
version: 1
connection: 1/12
IKE SA: created 1/19  established 1/19  times 0/2/10 ms
IPsec SA: created 1/27  established 1/27  times 0/1/10 ms
```

Listet alle IPsec SA auf ("phase2/tunnel up")

Nützliche CLI Kommandos - IPsec

```
[root:~]# ping 192.168.25.1   Ping vom Basler Lan zum Berner LAN
PING 192.168.25.1 (192.168.25.1): 56 data bytes
64 bytes from 192.168.25.1: icmp_seq=0 ttl=252 time=1.1 ms
64 bytes from 192.168.25.1: icmp_seq=1 ttl=253 time=0.6 ms
64 bytes from 192.168.25.1: icmp_seq=2 ttl=253 time=0.5 ms
64 bytes from 192.168.25.1: icmp_seq=3 ttl=253 time=0.3 ms
64 bytes from 192.168.25.1: icmp_seq=4 ttl=253 time=0.4 ms
---192.168.25.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.3/0.5/1.1 ms
FGT_BERN # get vpn ipsec tunnel summary
'Luzern_0' 198.18.0.5:0  selectors(total,up): 1/1  rx(pkt,err): 6/0  tx(pkt,err): 6/0
'Luzern' 198.18.0.2:0  selectors(total,up): 1/1  rx(pkt,err): 125/0  tx(pkt,err): 113/0
```

Nützliche CLI Kommandos - IPsec

```
FGT_BERN # diag vpn ike status detailed
vd: root/0
name: Luzern
version: 1
used-indices: 0-2
connection: 4/15
IKE SA: created 4/22  established 4/22  times 0/2/10 ms
IPsec SA: created 4/30  established 4/30  times 0/1/10 ms
FGT_BERN # diag netlink interface list
<...>
if=Luzernfamily=00 type=768 index=15 mtu=1438 link=0 master=0
ref=24 state=off start fw_flags=0 flags=up p2p run noarpmulticast
if=Luzern_0family=00 type=768 index=39 mtu=1438 link=15 master=0
ref=21 state=off start fw_flags=0 flags=up p2p run noarpmulticast
if=Luzern_1family=00 type=768 index=40 mtu=1438 link=15 master=0
ref=21 state=off start fw_flags=0 flags=up p2p run noarpmulticast
if=Luzern_2family=00 type=768 index=41 mtu=1438 link=15 master=0
ref=25 state=off start fw_flags=0 flags=up p2p run noarpmulticast
FGT_BERN # diag vpn ike gateway flush name Luzern_2
FGT_BERN # get vpn ipsec tunnel summary
'Luzern_0' 198.18.0.5:0  selectors(total,up): 1/1  rx(pkt,err): 8/0  tx(pkt,err): 10/0
'Luzern_1' 203.0.113.102:0  selectors(total,up): 1/1  rx(pkt,err): 7/0  tx(pkt,err): 8/0
'Luzern' 198.18.0.2:0  selectors(total,up): 1/1  rx(pkt,err): 144/0  tx(pkt,err): 130/0
```

Fährt einen Shortcut herunter

Shortcuts können nicht über das WebGui geflusht werden, nur über die CLI

Nützliche CLI Kommandos - IPsec

Ab FortiOS 6.0 können mehrere IP-Adressen im IKE Debug Filter angegeben werden (**m**dst-addr4)
Es vereinfacht das Debugging von Spoke-to-Spoke Negationen:

```
# Vom Spoke-A, die negotiation mit Spoke-B überprüfen (welche initial durch
den Hub geht)
diag debug console timestamp enable
diag vpn ike log filter clear
diag vpn ike log filter mdst-addr4 <ip.vom.Hub> <ip.vom.Spoke-B>
diag debug application ike-1
diag debug application fnbamd-1# nur wenn Zertifikat Auth. benutzt wird.
diag debug enable
```

Nützliche Kommandos - Routing

```
FGT_BERN # get router info bgp summary
BGP router identifier 10.10.10.2, local AS number 65000
BGP table version is 2
2 BGP AS-PATH entries
0 BGP community entries
Neighbor        V         AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down  State/PfxRcd
10.10.10.1      4      65000    2061    2049      2   0    0 00:00:23        4
Total number of neighbors 1
FGT_BERN # get router info bgpnetwork
BGP table version is 2, local router ID is 10.10.10.2
Status codes: s suppressed, d damped, h history, * valid, > best, i-internal,
              S Stale
Origin codes: i-IGP, e -EGP, ? -incomplete
Network             Next Hop            Metric LocPrf Weight Path
*>i192.168.1.0      10.10.10.1               0    100      0 i
*> 192.168.2.0      0.0.0.0                       100  32768 i
*>i192.168.25.0     10.10.10.3               0    100      0 i
*>i192.168.101.0    10.255.255.2             0    100      0 65100 i
*>i192.168.102.0    10.20.20.2               0    100      0 65100 i

Total number of prefixes 5
```

BGP Peers

BGP Table

Nützliche Kommandos - Routing

```
FGT_BERN # get router info bgp network 192.168.102.0
BGP routing table entry for 192.168.102.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Not advertised to any peer
65100
10.20.20.2 from 10.10.10.1 (10.10.10.1)
Origin IGP metric 0, localpref100, valid, internal, best
Last update: Tue Jun 21 17:03:02 2016

FGT_BERN # get router info routing-table bgp
B       192.168.1.0/24 [200/0] via 10.10.10.1, Luzern, 00:01:03
B       192.168.3.0/24 [200/0] via 10.10.10.3 (recursive via 10.10.10.1), 00:01:03
B       192.168.101.0/24 [200/0] via 10.255.255.2 (recursive via 10.10.10.1), 00:01:03
B       192.168.102.0/24 [200/0] via 10.20.20.2 (recursive via 10.10.10.1), 00:01:03
FGT_BERN # get router info routing-table static

S*      0.0.0.0/0 [10/0] via 198.51.100.254, port2
S       10.10.10.0/24 [10/0] via 10.10.10.1, Luzern
S       10.20.20.0/24 [10/0] via 10.10.10.1, Luzern
S       10.255.255.0/30 [10/0] via 10.10.10.1, Luzern

FGT_BERN # get router info routing-table connected
C       10.5.48.0/20   is directly connected, port10
C       10.10.10.1/32  is directly connected, Luzern
C       10.10.10.2/32  is directly connected, Luzern
C       192.168.2.0/24 is directly connected, port1
C       198.18.0.0/27  is directly connected, port2
```

BGP Details von bestimmten Präfixen

BGP Routen in der RIB

Statische Route in der RIB

Verbunde Routen in der RIB

Nützliche Kommandos - Routing

```
FGT_BERN # get router info routing-table all
Codes: K -kernel, C -connected, S -static, R -RIP, B -BGP
O -OSPF, IA -OSPF inter area
N1 -OSPF NSSA external type 1, N2 -OSPF NSSA external type 2
E1 -OSPF external type 1, E2 -OSPF external type 2
i-IS-IS, L1 -IS-IS level-1, L2 -IS-IS level-2, ia-IS-IS inter area
* -candidate default
S*      0.0.0.0/0 [10/0] via 198.51.100.254, port2
C       10.5.48.0/20 is directly connected, port10
S       10.10.10.0/24 [10/0] via 10.10.10.1, Luzern
C       10.10.10.1/32 is directly connected, Luzern
C       10.10.10.2/32 is directly connected, Luzern
S       10.20.20.0/24 [10/0] via 10.10.10.1, Luzern
R       10.255.255.0/24 [120/2] via 10.10.10.1, Luzern, 00:02:14
S       10.255.255.0/30 [10/0] via 10.10.10.1, Luzern
B       192.168.1.0/24 [200/0] via 10.10.10.1, Luzern, 00:01:08
C       192.168.2.0/24 is directly connected, port1
B       192.168.3.0/24 [200/0] via 10.10.10.3 (recursive via 10.10.10.1), 00:01:08
B       192.168.101.0/24 [200/0] via 10.255.255.2 (recursive via 10.10.10.1), 00:01:08
B       192.168.102.0/24 [200/0] via 10.20.20.2 (recursive via 10.10.10.1), 00:01:08
C       198.51.100.0/24 is directly connected, port2

FGT_BERN # get router info routing-table details 192.168.102.1
Routing entry for 192.168.102.0/24
Known via "bgp", distance 200, metric 0, best
Last update 00:02:48 ago
* 10.20.20.2 (recursive via 10.10.10.1)
```

Alle aktiven Routen in dem RIB

Spezifische Details einer Route in dem RIB

Nützliche Kommandos - Routing

```
[root:~]# ping 192.168.3.1
[root:~]# ping 192.168.101.1
[root:~]# ping 192.168.102.1
```

Aktivieren eines Shortcuts

```
FGT_BERN (root) # get router info routing-table all
Codes: K -kernel, C -connected, S -static, R -RIP, B -BGP
O -OSPF, IA -OSPF inter area
N1 -OSPF NSSA external type 1, N2 -OSPF NSSA external type 2
E1 -OSPF external type 1, E2 -OSPF external type 2
i-IS-IS, L1 -IS-IS level-1, L2 -IS-IS level-2, ia-IS-IS inter area
* -candidate default
S*      0.0.0.0/0 [10/0] via 198.51.100.254, port2
C       10.5.48.0/20 is directly connected, port10
S       10.10.10.0/24 [10/0] via 10.10.10.1, Luzern
C       10.10.10.1/32 is directly connected, Luzern
C       10.10.10.2/32 is directly connected, Luzern
                      is directly connected, Luzern_0
                      is directly connected, Luzern_1
                      is directly connected, Luzern_2
C       10.10.10.3/32 is directly connected, Luzern_0
S       10.20.20.0/24 [10/0] via 10.10.10.1, Luzern
C       10.20.20.2/32 is directly connected, Luzern_2
S       10.255.255.0/24 [10/0] via 10.10.10.1, Luzern
C       10.255.255.2/32 is directly connected, Luzern_1
B       192.168.1.0/24 [200/0] via 10.10.10.1, Luzern, 00:20:39
C       192.168.2.0/24 is directly connected, port1
B       192.168.3.0/24 [200/0] via 10.10.10.3, Luzern_0, 00:01:11
B       192.168.101.0/24 [200/0] via 10.255.255.2, Luzern_1, 00:00:11
B       192.168.102.0/24 [200/0] via 10.20.20.2, Luzern_2, 00:00:11
C       198.51.100.0/24 is directly connected, port2
```

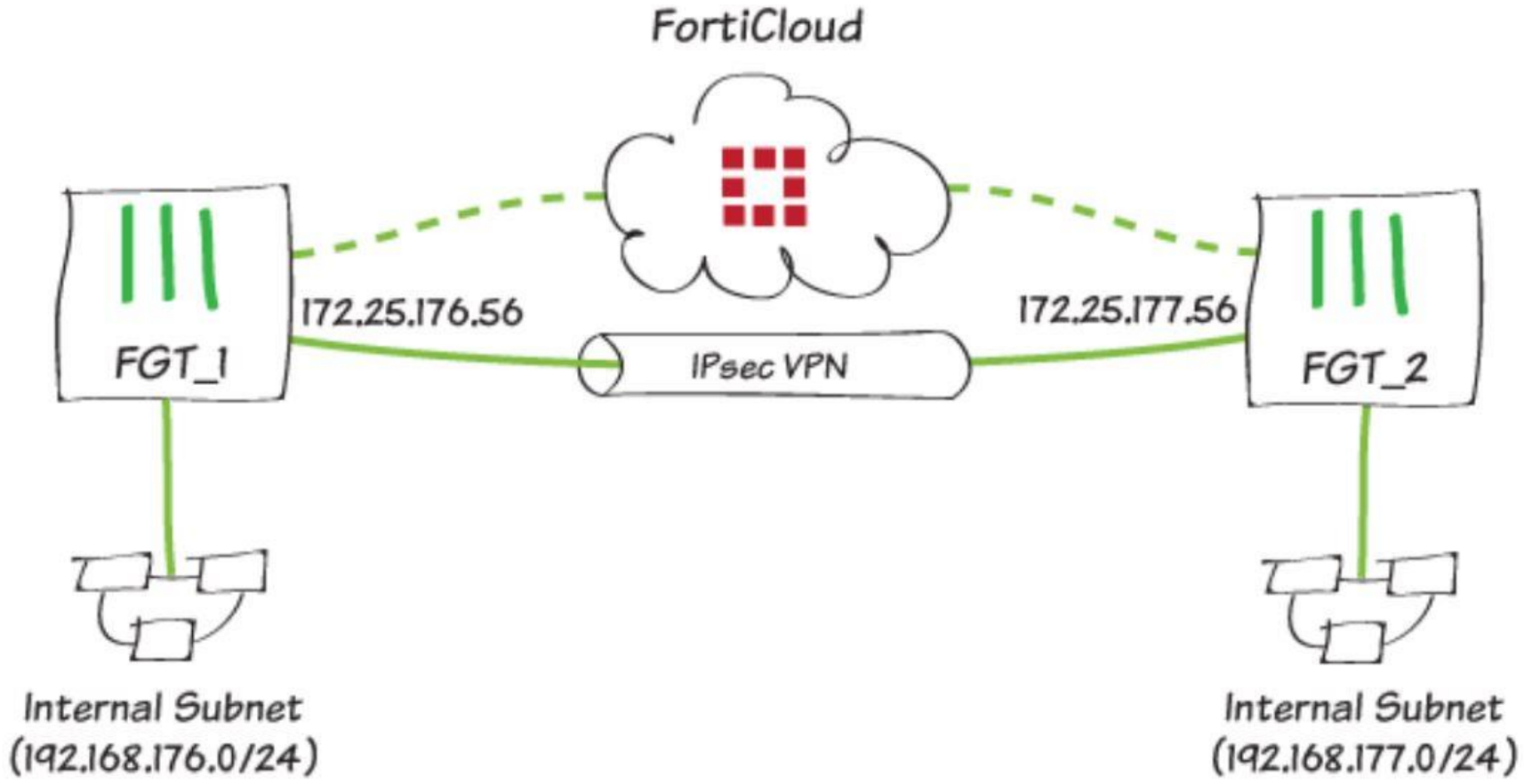
Overlay `local-ip`

BGP Next-Hop welche direkt verbunden sind

Routen durch den Shortcut

One-Click VPN (OCVPN)

Was ist OCVPN

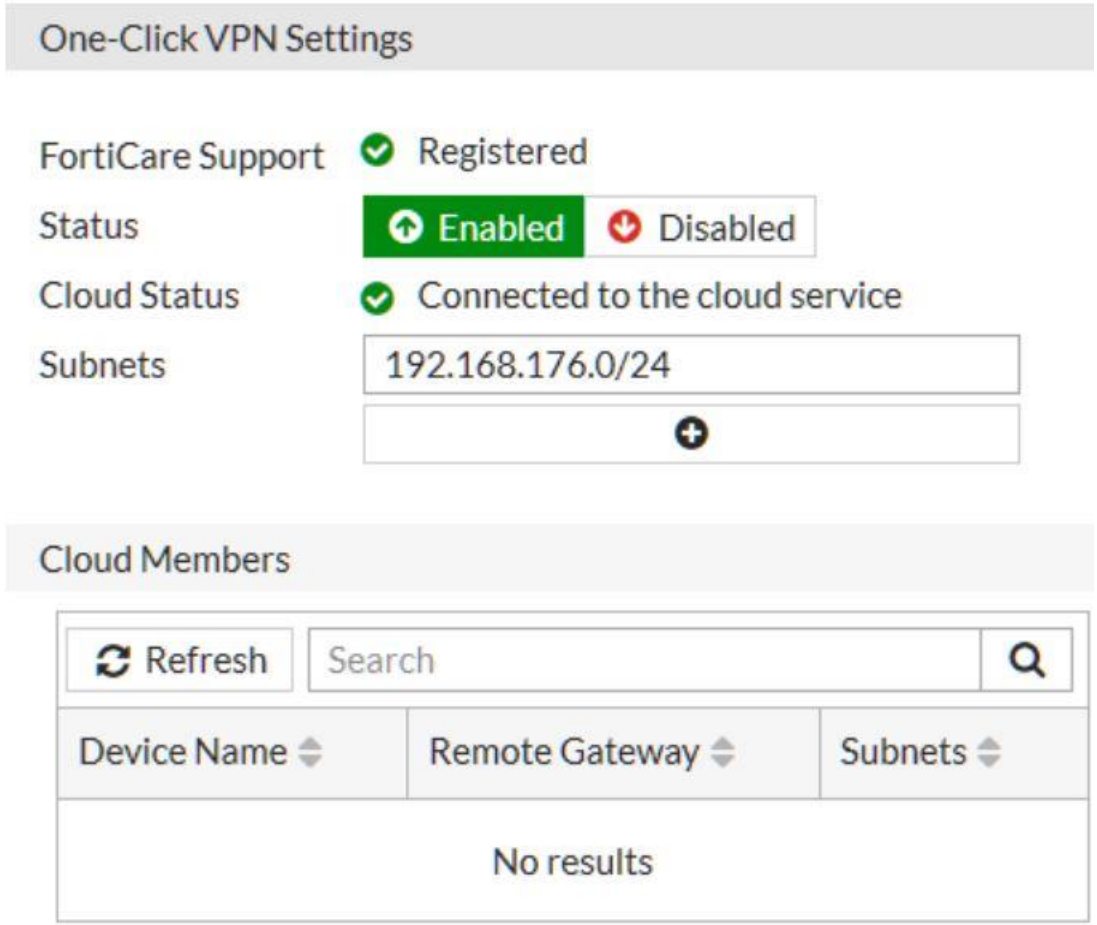


Limitierungen vom OCVPN

- Die FortiGate muss mit einer gültigen FortiCare Supportlizenz registriert sein.
- Es werden nur Full-Mesh-VPN-Konfigurationen mit PSK-Key unterstützt.
- Es müssen öffentliche IPs verwendet werden (FortiGates hinter NAT-Devices funktionieren nicht.)
- Nicht-root VDOMs und FortiGate VMs werden nicht unterstützt.
- In der OCVPN-Cloud können bis zu 16 Devices mit jeweils maximal 16 Subnetzen hinzugefügt werden.

OCVPN aktivieren – FGT_1

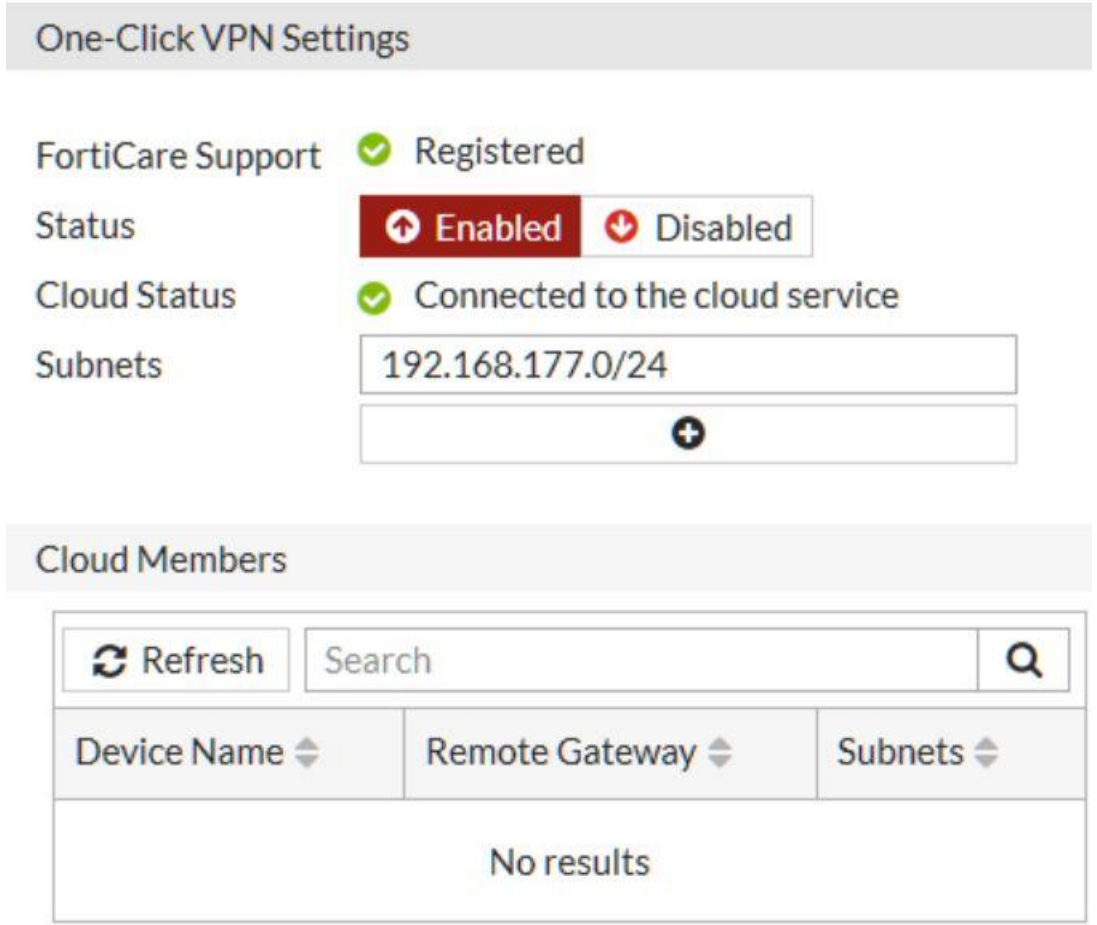
- Menu **VPN > One-Click VPN Settings**.
- Status auf **Enabled** setzen und Cloud-Status abwarten. Dies kann ein oder zwei Minuten dauern.
- Wie angegeben, erscheint ein grünes Häkchen zusammen mit der Meldung Verbunden mit dem Cloud-Service.
- Die Subnetze von der **FGT_1** hinzufügen



OCVPN aktivieren FGT_2

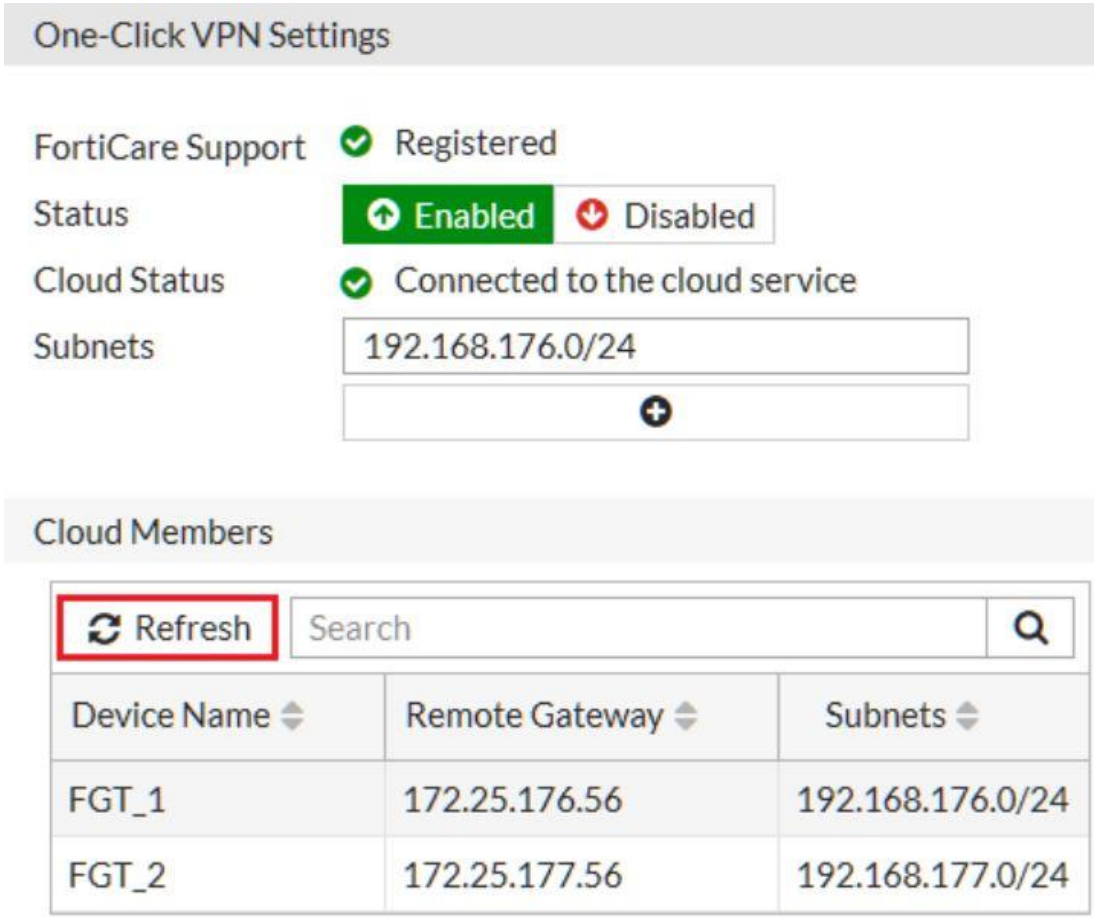
- Auf der **FGT_2** ist das vorgehen gleich wie vorher.

- Die dazugehörigen Subnetze der **FGT_2** hinzufügen.



Bestätigung der Cloud-Mitgliedschaft

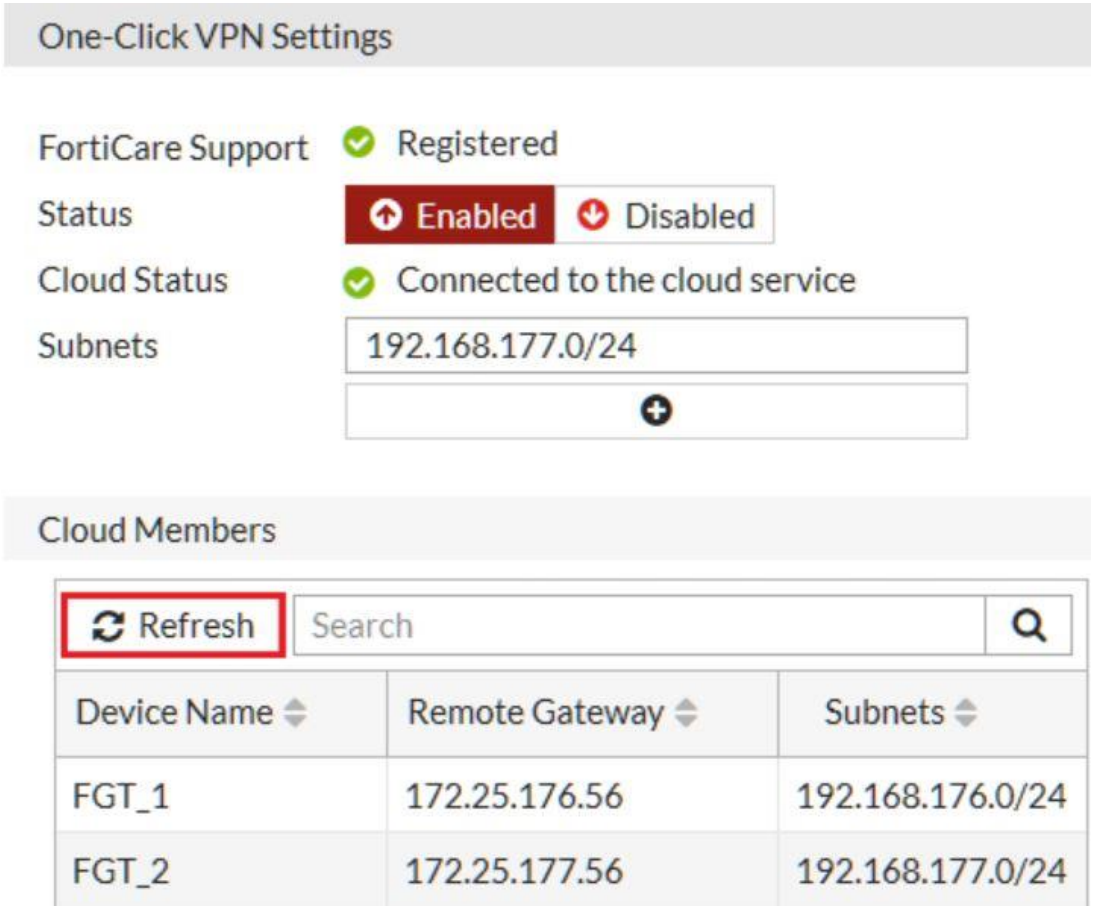
- In der Tabelle Cloud Members FGT_1 anklicken und **Refresh** anwählen.
- Die Remote Gateways und die entsprechenden Subnetze für jedes Devices sollten in der Liste erscheinen.



Bestätigung der Cloud-Mitgliedschaft

- Der vorgängige Schritt für jede FortiGate ausführen welche auch Mitglied in der OCVPN Cloud ist.

- FGT_2 sollte in der Tabelle die selben Resultate liefern wie vorher bei der FGT_1 Firewall.

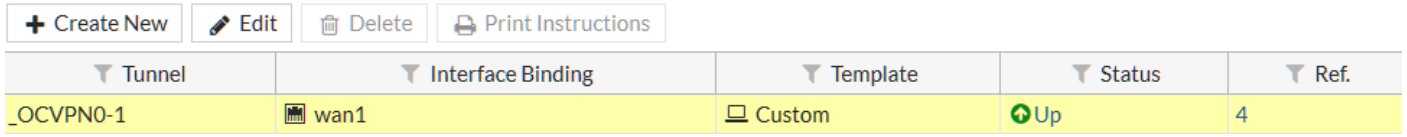


Resultat

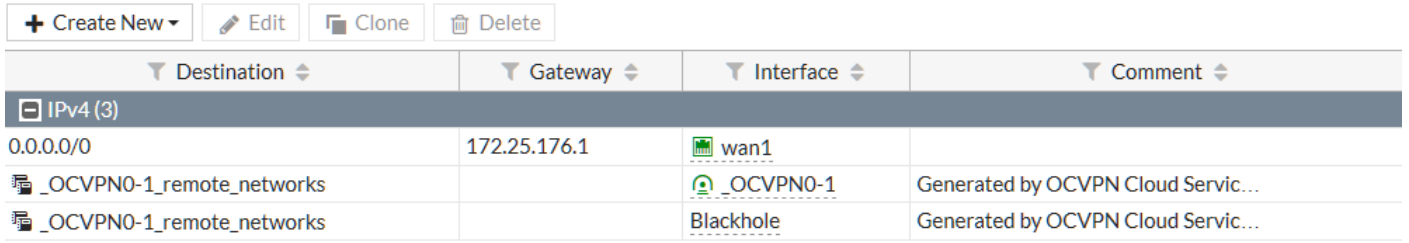
- Während die Tabelle der Cloud-Mitglieder ergänzt wird, aktualisiert die OCVPN-Cloud jedes Mitglied automatisch.
- Jetzt kann der Rest der Konfiguration überprüft werden und der Tunnel überprüft werden, ob er funktioniert.

Überprüfen der VPN Konfiguration

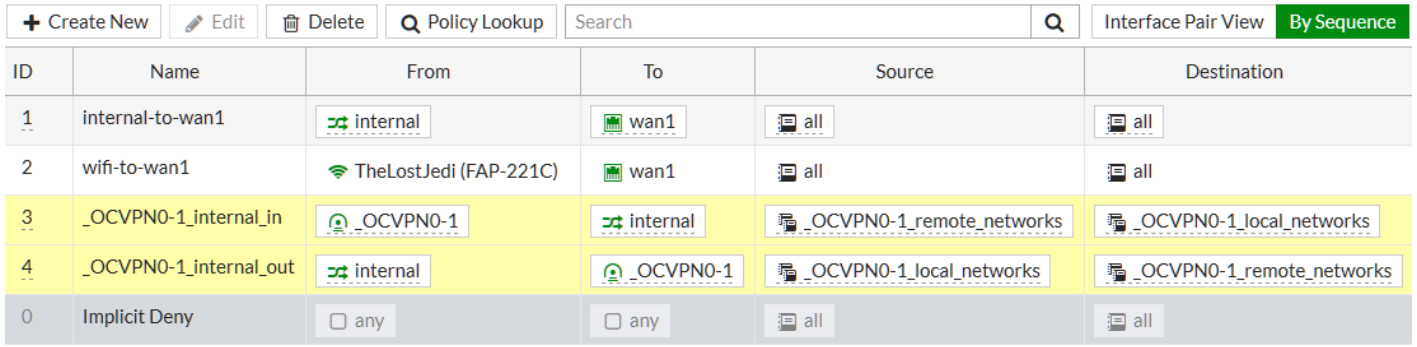
Auf den beiden FortiGate auf **VPN -> Ipsec Tunnels** und überprüfen ob der neue Tunnel mit dem Präfix _OCVPN vorhanden ist.



Unter **Network -> Static Routes** die neu angelegten Routen überprüfen.

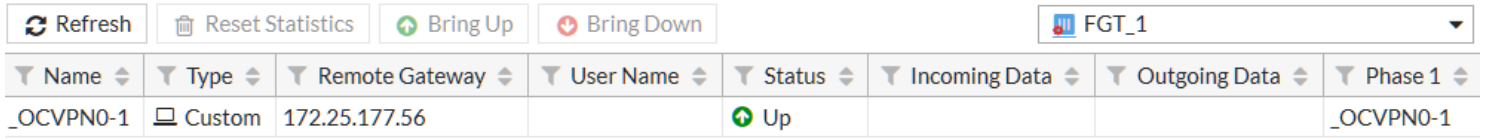


Unter **Policy & Objects -> IPv4 Policy** die neuen Regeln überprüfen

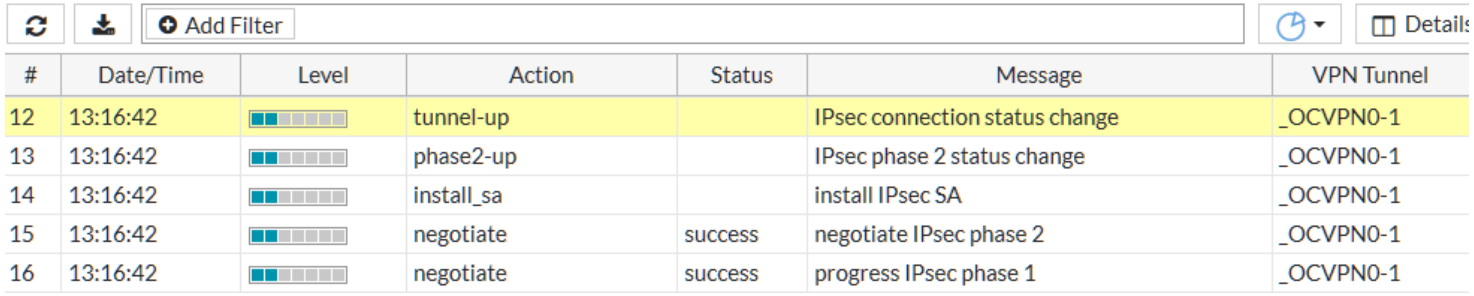


Überprüfen der VPN Konfiguration

Im Menu **Monitor -> Ipsec Monitor** kann verifiziert werden, ob der Status des Tunnels up ist.



Unter **Log & Report -> VPN Events** kann die Tunnelstatistik angeschaut werden.



Dokumentation

- https://cookbook.fortinet.com/ocvpn-60/
- http://help.fortinet.com/fos60hlp/60/Content/FortiOS/fortigate-ipsecvpn/OCVPN/ocvpn_intro.htm
- https://www.youtube.com/watch?v=g7mrEXq8D24 (Video)